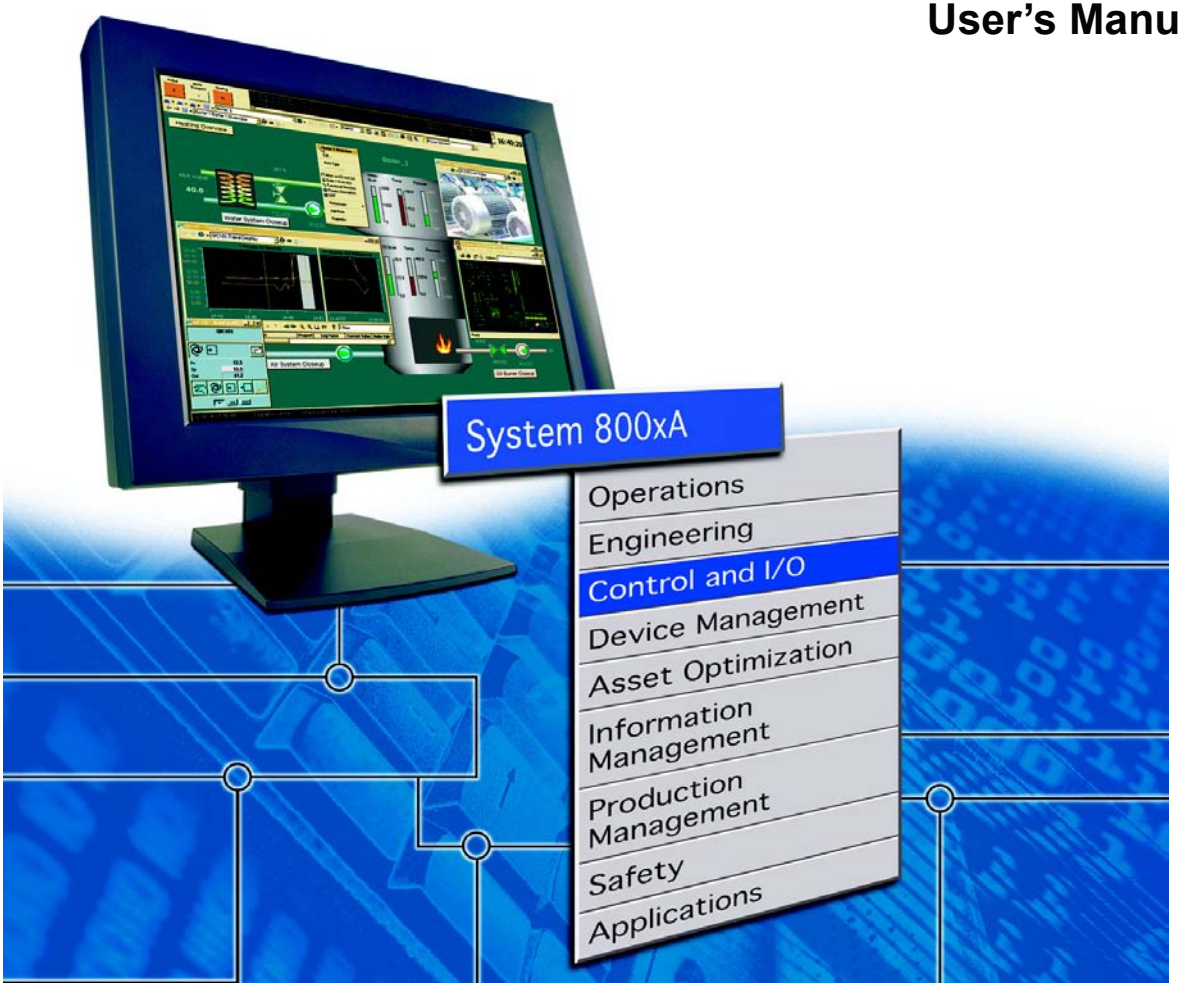
# Industrial IT
## 800xA - Control and I/O
### System Version 4.0

# CI858 DriveBus Communication Interface
## User's Manual




Industrial IT enabled

ABB

# Industrial$^{IT}$
## 800xA - Control and I/O
### System Version 4.0

# CI858 DriveBus Communication Interface
## User's Manual

## NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without written permission from ABB, and the contents thereof must not be imparted to a third party nor used for any unauthorized purpose.

The software or hardware described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

This product meets the requirements specified in EMC Directive 89/336/EEC and in Low Voltage Directive 72/23/EEC.

Copyright © 2003 - 2004 by ABB.
All rights reserved.

Release: October 2004
Document number: 3AFE68237432 Rev C

## TRADEMARKS

Registrations and trademarks used in this document include:

| | |
|---|---|
| Windows | Registered trademark of Microsoft Corporation. |
| ActiveX | Registered trademark of Microsoft Corporation. |
| PostScript | Registered trademark of Adobe Systems Inc. |
| Acrobat Reader | Registered trademark of Adobe Systems Inc. |
| Industrial IT | Trademark of ABB. |

# TABLE OF CONTENTS


## About This Book

Introduction ..... 9
Terminology ..... 10
Related Documents ..... 12

## Safety Summary

Introduction ..... 13
General Safety Instructions for AC and DC Drives ..... 13

## Section 1 - Hardware and Connections

Introduction ..... 15
CI858 Features ..... 17
CI858 Connections ..... 19
- Drive Channel / DDCS DriveBus ..... 20
- Special I/O Channel ..... 22
- PC Tool Channel ..... 23

## Section 2 - Mounting

Introduction ..... 25
- Mounting the Unit onto the DIN rail ..... 25

## Section 3 - Communication

Introduction ..... 27
Dataset Communication ..... 29
- Example of Datasets in the ACS800 Standard Application Program ..... 30
- Example of Datasets in the ACS800 System Application Program ..... 31

## Section 4 - Configuration

Introduction ..... 33

Control Builder ..... 33
CI858 Configuration ..... 36
Adding a CI858 Unit ..... 36
CI858 Parameters ..... 36
Drive Configuration ..... 38
Adding a Drive ..... 38
Drive Parameters ..... 39
ABB Standard Drive and ABB Engineered Drive ..... 39
ABB Drive Template (basic/extension) ..... 40
BusManager ..... 41
Special I/O Configuration ..... 42
Adding a Special I/O Unit ..... 42
Special I/O Parameters ..... 43
Special I/O Template ..... 44
Connections ..... 44
Special I/O Function Blocks ..... 46
How to Insert a Library File ..... 46
How to Insert a Special I/O Unit ..... 46
Activating a Function Block ..... 47
NBIO-21 ..... 48
BIO2CONF ..... 48
BIO2DI ..... 50
BIO2DO ..... 50
BIO2AO ..... 51
BIO2AI ..... 54
NBIO-31 ..... 56
BIO3DI ..... 56
BIO3DO ..... 57
NDIO ..... 58
NDIO1DI ..... 58
NDIO1DO ..... 59
NAIO ..... 60

NAIO2AO ..... 60
NAIO2AI ..... 63
NPCT ..... 65
PCT_CONF ..... 65
SPEEDPO1 ..... 69
SPEED_R ..... 73
POS_IL1 ..... 74
CONV24 ..... 75
NTAC ..... 76
TAC_SP ..... 77
NWIO ..... 78
NWIO1DI ..... 79
NWIO1DO ..... 80
NCTI ..... 82
NDSC ..... 82
Output Error ..... 82
Changing the CI858 System Settings ..... 83
Online Help ..... 83

## Appendix A - Technical Data

General Technical Data ..... 85
Ambient Data ..... 85
LED Indicators ..... 86
Power Supply Requirements ..... 87
Optical Connectors ..... 87
DIN rail - TS 35/7.5 ..... 88
CEB Connector, X1 ..... 89
CEX Bus Interface ..... 91
Firmware Download ..... 91

# About This Book

## Introduction

This manual introduces the installation and start-up of the CI858 communication interface.

CI858 is a communication interface for the DriveBus protocol. ABB Drives and Special I/O units communicate with the AC 800M controller via the CI858. The CI858 is designed especially for sectional drive applications, for example ABB rolling mill drive systems and ABB paper machine drive systems.

***Section 1, Hardware and Connections***

Provides an overview of the CI858 hardware and physical connections.

***Section 2, Mounting***

Describes the mounting of the CI858.

***Section 3, Communication***

Introduces the software and the content of messages used for communication between the CI858, AC 800M, ABB Drives and Special I/O units.

***Section 4, Configuration***

Describes how to prepare the CI858 and its units for communication i.e. how to make logical connections between the AC 800M application variables and the ABB Drives' internal variables

***Appendix A, Technical Data***

Contains technical data.

# Terminology

The following table lists the terms used in this document.

| Term | Description |
|---|---|
| ABB Drives | DC and AC drives by ABB |
| ABB Drive Template (basic / extension) | ABB Drive Template (basic) is a CI858 configuration option with eight dataset pairs.With ABB Drive Template (extension) the dataset pair number can be extended |
| ABB Standard Drive | CI858 configuration option with two dataset pairs. Standard Drive option is used with Standard Application firmware. |
| ABB Engineered Drive | CI858 configuration option with eight dataset pairs. Engineered Drive option is used with System Application firmware. |
| AC 800M | Advant Controller 800M series, general purpose process controller |
| CEB | Communication Expansion Base Plate |
| CEM | Communication Expansion Module |
| CEX bus | Communication Expansion Bus |
| CI858 | Communication Interface |
| Control Builder | engineering tool |
| CPU | Central processing unit |
| CSR | Communication and Status Register |
| DDCS | Distributed Drives Communication System |
| DPM | Dual port memory |
| DPRAM | Dual Port Random Access Memory |

| Term | Description |
|---|---|
| DriveBus | Communication link dedicated for ABB drives |
| DS | Dataset |
| DSP | Digital Signal Processor |
| DW | Data word |
| EMC | Electromagnetic Compatibility |
| FPGA | Field Programmable Gate Array |
| hardware tree | The data structure of a hardware configuration for a controller. |
| ICMC02 | Integrated Control module Circuit (=communications ASIC) |
| LED | Light Emitting Diode |
| LVDS | Low Voltage Differential Signaling |
| NAIO | Analogue I/O Extension Module |
| NBIO | Basic I/O Unit |
| NCTI | Crane Transducer Interface |
| NDBU | Branching Unit |
| NDIO | Digital I/O Extension Module |
| NDSC-01 | control board of the ACS600 Diode Supply Unit |
| NPCT | Pulse Counter and Timer Unit |
| NTAC | Pulse Encoder Interface Module |
| NWIO | Watchdog I/O Module |
| ONCE | On-Chip Emulation unit |
| PPP | Point-to-Point Protocol |
| TCP/IP | Transmission Control Protocol/Internet Protocol |

# Related Documents

The following table lists the documentation related to this manual.

| Category | Title |
|---|---|
| Hardware | AC 800M Controller Hardware, [3BSE 027 941(English)] |
| | NDBU-85/95 DDCS Branching Units User's Manual, [3BFE 64285513(English)] |
| | NTAC-0x/NDIO-0x/NAIO-0x Installation and Start-up Guide [3AFY 58919730 (English)] |
| | Crane transducer Interface User's manual [3ASC23F207 (English)] |
| | ACA 631/633 Diode Supply Section User's Manual [3AFY 61451544 (English)] |
| | NPCT-01 Pulse Counter/Timer Unit description [3AFY64362828(English)] |
| Software | AC 800M/C Communication, Protocols and Design, [3BSE 028 811(English)] |
| | AC 800M Firmware Delivery CD-ROM |

# Safety Summary

## Introduction

The complete safety instructions stated in the appropriate ABB Drive *Hardware Manual* must be followed when installing, operating and servicing the frequency converters. Study the complete safety instructions carefully.

## General Safety Instructions for AC and DC Drives

These general safety instructions include only main parts of the complete safety instructions. Neglecting these instructions may cause physical injury or death.

**WARNING!** All electrical installation and maintenance work on the drive should be carried out by qualified electricians.

Ground the drive, the motor and adjoining equipment to ensure personnel safety in all circumstances, and to reduce electromagnetic emission and pick-up.

Any installation work must be done with power off, and power is not to be reconnected unless the installation work is complete. Wait 5 minutes after switching off the supply before starting work. Always ensure by measuring that the drive is de-energized.

If the auxiliary voltage circuit of the drive is powered from an external power supply, opening the disconnecting device does not remove all voltages. Before starting work, check which circuits remain live after opening of the disconnecting device by referring to the circuit diagrams for your particular delivery. Ensure by measuring that the part of the cabinet you are working on is not live.

Do not make any voltage withstand tests on any part of the unit while the unit is connected.

Disconnect motor cables before making any measurements on motors or motor cables.

Live parts on the inside of doors are protected against direct contact. Special safety attention shall be paid when handling shrouds made of sheet metal.

Check the cable connections at the shipping split joints before switching on the supply voltage.

**WARNING!** Close switch fuses of all parallel connected inverters before starting the frequency converter.

Do not open the drive section switch fuses when the inverter is running.

**WARNING!** Fans may continue to rotate for a while after the disconnection of the electrical supply.

**WARNING!** Some parts like heatsinks of power semiconductors and toroidal cores on motor cables inside of cabinet remain hot for a while after the disconnection of the electrical supply.

# Section 1 Hardware and Connections

## Introduction

The CI858 unit consists of a communication expansion base plate board CEB, a communication expansion module board CEM, and mechanics. All electronic devices and all functions are located on the CEM board.

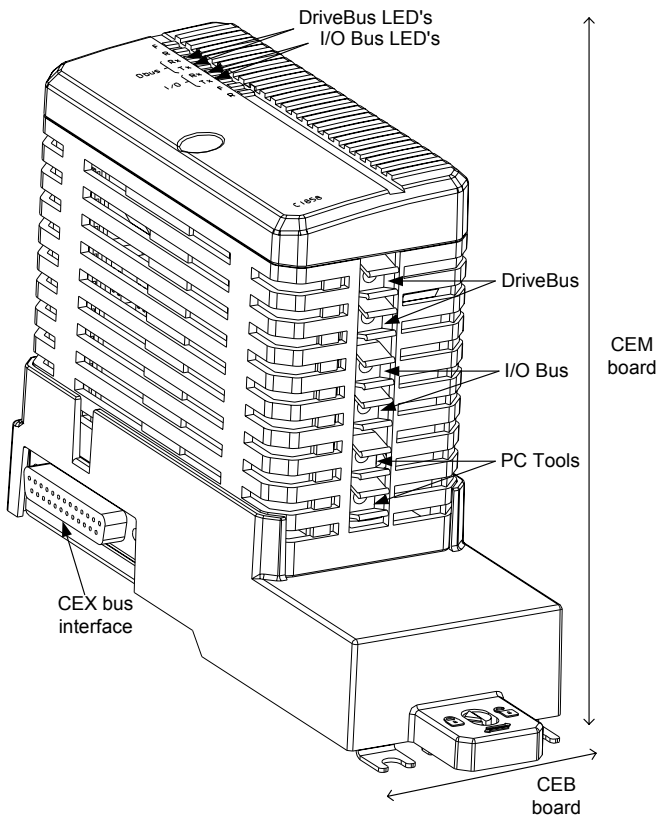



Figure 1. CI858 communication interface.

The CEM board connects via connector X1 to the CEB board, which in turn is connected to the CEX bus.

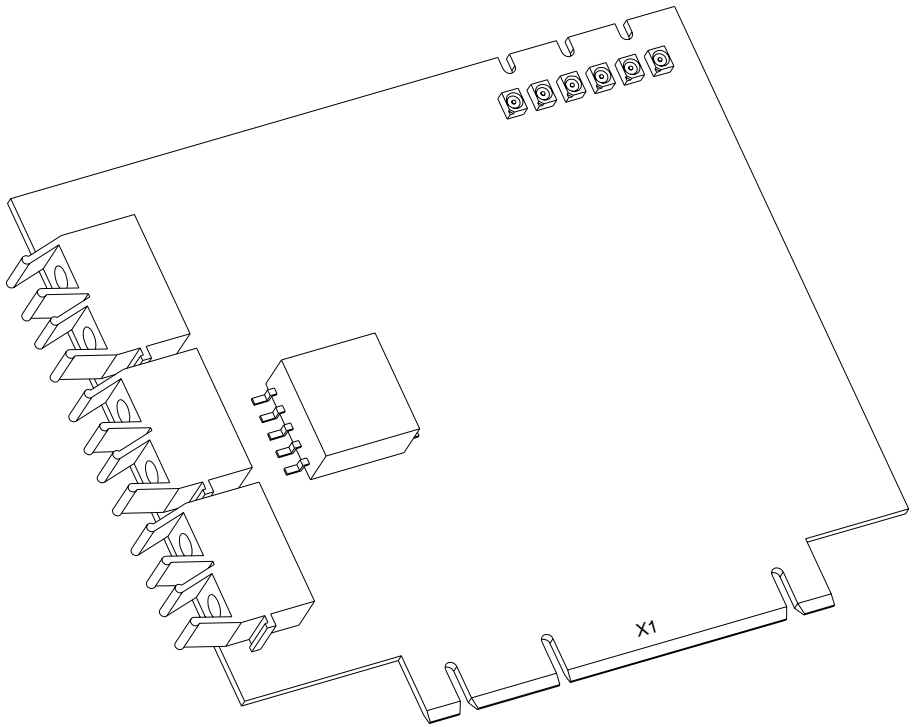



Figure 2. CI858 CEM board.

# CI858 Features

The maximum number of CI858 units connected to the AC 800M is two.

The functional structure of the CI858 includes

- CEX bus interface:

  Communication between the CI858 and the AC 800M is implemented through the CEX bus. The CEX bus operates with a clock frequency of 64 MHz. The maximum transfer data is ~1.25 MBytes/s.

- DDCS interfaces for
  - DriveBus
  - I/O Bus
  - PC Tool
- +24 V power supply from the CEX bus.
- DC/DC conversion from +24 V to +5 V, + 3.3V and +2.5 V.
- Motorola Digital Signal processor DSP56303 with 72 MHz CPU frequency.
- Xilinx XC2S50 FPGA for the CEX bus interface. 94 user I/O pins available on CI858.
- 32 kB boot PROM, AT27LV256, for boot code.
- 1 MB flash PROM, Am29DL800B, for program code.
- 768 kB SRAM program and data memory.

AC 800M and CI858 communicate with each other via dual-port RAM, DPRAM. The DPRAM has the following functionalities:

- Dual-port memory, 32 * 16 kB.
- One interrupt in each direction.
- Eight semaphores which simplifies the utilization of shared resources.
- ONCE port which can be used for reprogramming the module.

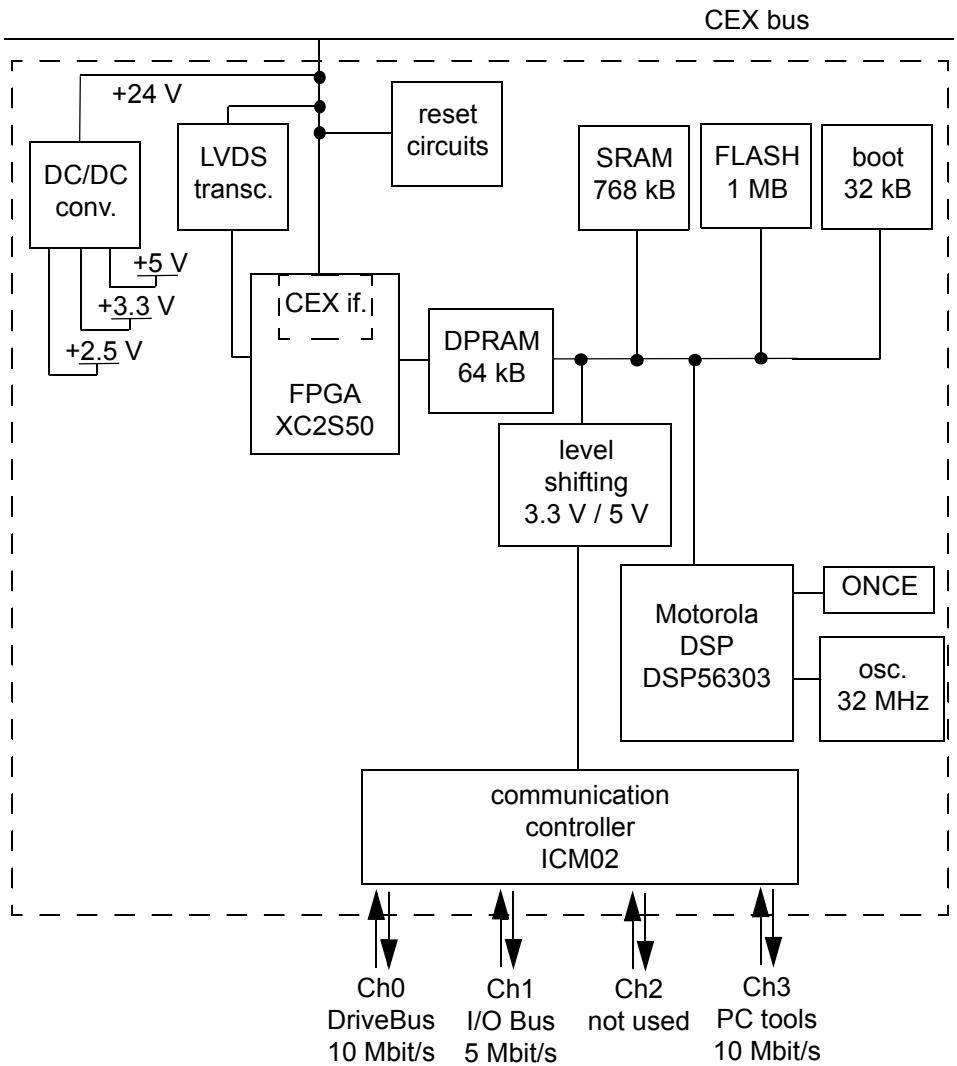



Figure 3. Functional block diagram of the CI858.

# CI858 Connections

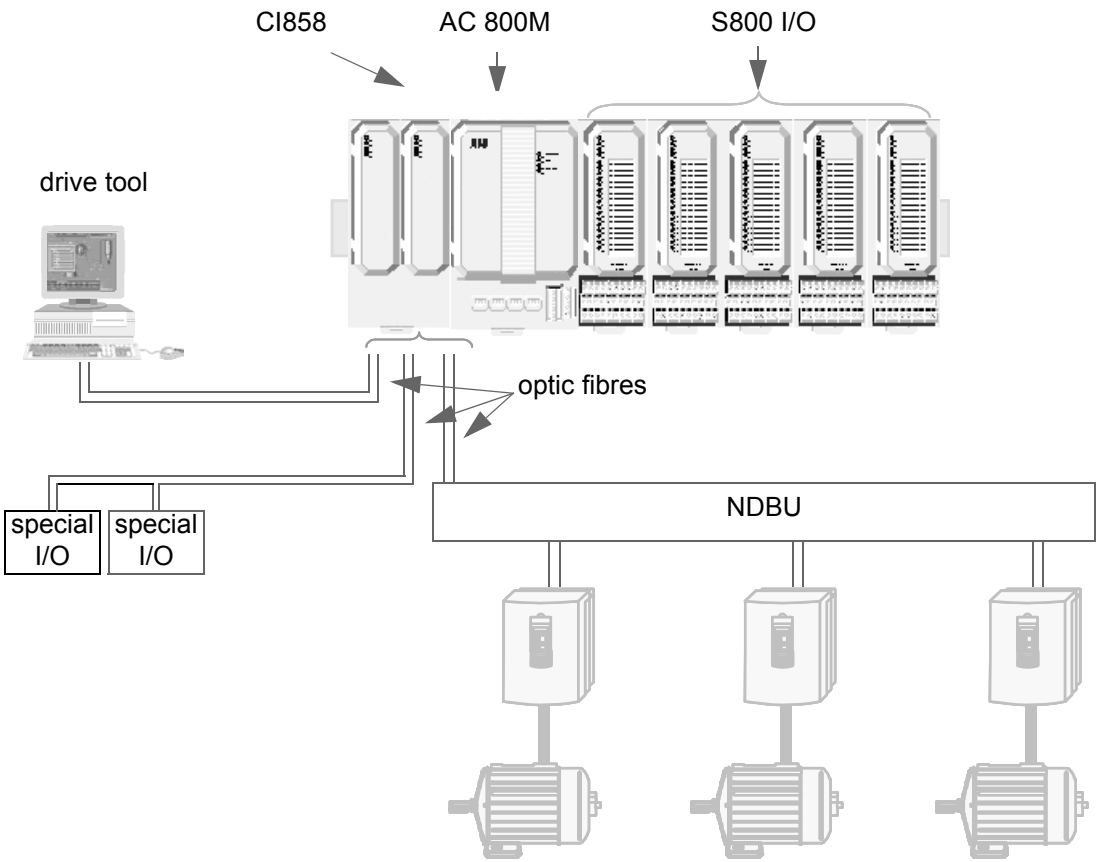



Figure 4. CI858 connections.

## Drive Channel / DDCS DriveBus

ABB Drives can be connected to the AC 800M via the CI858.

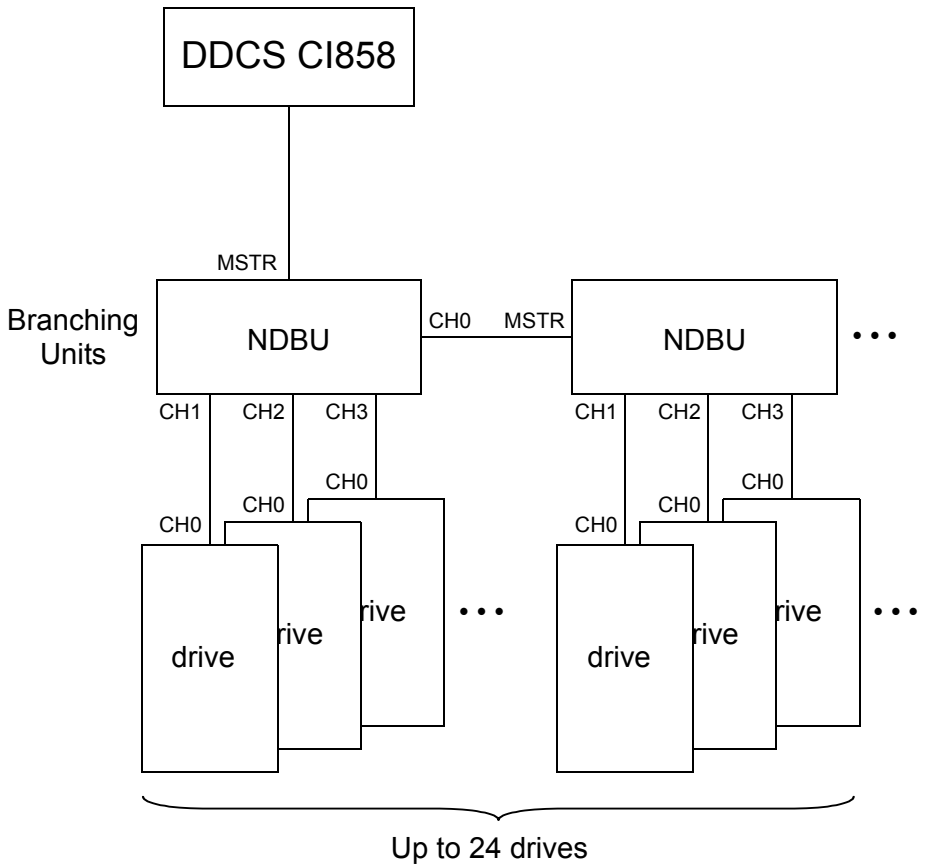



Figure 5. DriveBus topology.

Main communication functions of the Drive channel are:

- dataset communication
- broadcast system time
- supervision and diagnostics functions

The data exchange between the AC 800M and ABB Drives consists of dataset pairs. The DriveBus is able to transfer at the maximum 8 dataset pairs / 1 ms.

The Drive channel can be used for controlling up to 24 drives. Connecting more than one drive to the CI858 requires the use of a branching unit NDBU, which enables the construction of a logical bus with physical star topology. The branching units can be chained. For more information on the NDBU branching unit, see *DDCS Branching Units User's Manual, 3BFE 64285513.*

The Drive channel supports the following drives:

- ACS800 / ACS600 SingleDrive
- ACS800 / ACS600 MultiDrive
- ACS800 / ACS600 IGBT supply units
- ACS600 thyristor supply units
- ACS140...ACS400
- DCS600 and DCS400
- ACS6000 product family / large drives
- ACS1000 product family
- future drive types which are provided with DDCS interface
- special drive applications, see ABB Drive Templates in *Drive Parameters* on page 39

## Special I/O Channel

**Note:** Special I/O will not be a part of the Control and I/O functionality in the System version 3.1 released and owned by ABB Automation Technology Products AB.

The CI858 Special I/O channel can be used to connect up to 12 I/O's per unit. The I/O channel is able to transfer 2 dataset pairs / 1 ms. The I/O units are connected in a ring topology.

The following I/O units are supported:

- NAIO-03 Analogue I/O Extension Module - contains 2 current or voltage inputs and 2 current outputs
- NBIO-21 Basic I/O Unit 2 - contains 3 digital inputs, 2 analogue inputs, 2 digital outputs and 2 analogue outputs
- NBIO-31 Basic I/O Unit 3 - contains 4 digital inputs and 3 digital outputs
- NCTI-01 Crane Transducer Interface
- NDIO-02 Digital I/O Extension Module - contains 2 digital inputs and 2 relay outputs
- NDSC-01 control board of the ACS600 Diode Supply
- NPCT-01 Pulse Counter and Timer Unit - contains 2 encoder inputs, 4 digital inputs, 4 digital outputs
- NTAC-02 Pulse Encoder Interface Module - contains an interface for a digital pulse encoder connection
- NWIO-01 Watchdog I/O Module - contains watchdog logic for supervising the operation of the software
- special I/O applications, see *Special I/O Parameters* on page 43

## PC Tool Channel

The PC Tool channel can be used for downloading firmware to the CI858. The firmware is downloaded with a special loading package and it does not involve using the Control Builder engineering tool. For instructions on how to download CI858 firmware, see AC 800M Firmware Delivery CD-ROM: DELIVERYFIRMWARE\FIRMWARE\CI858.

# Section 2 Mounting

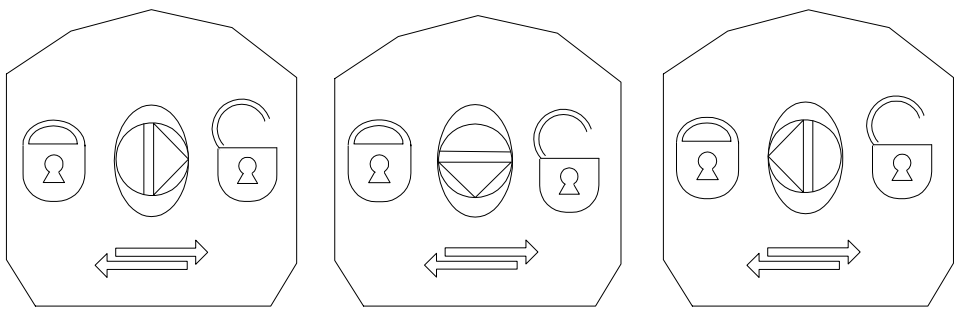
## Introduction

The CI858 is powered from the processor unit via the CEX bus and requires therefore no additional external power source. The unit is mounted onto a horizontal DIN rail.

### Mounting the Unit onto the DIN rail

The base plate has a locking mechanism that connects the metallic back plate to the DIN rail, providing an effective ground connection.

1. Rotate the locking device to the OPEN position. Hook the upper lip at the rear of the base plate over the upper edge of the DIN rail and snap the base plate fully into the mounting position.
2. Rotate the locking device to the SLIDE position. Slide the unit along the DIN rail to the desired mounting position and attach it to the adjacent unit base plate with the connector plugs and sockets.
3. Rotate the locking device clockwise to the LOCKED position.



The additional screw lugs, located in the lower part of the base plate, serve no electrical function. They provide secure mounting in locations subject to vibration.

# Section 3 Communication

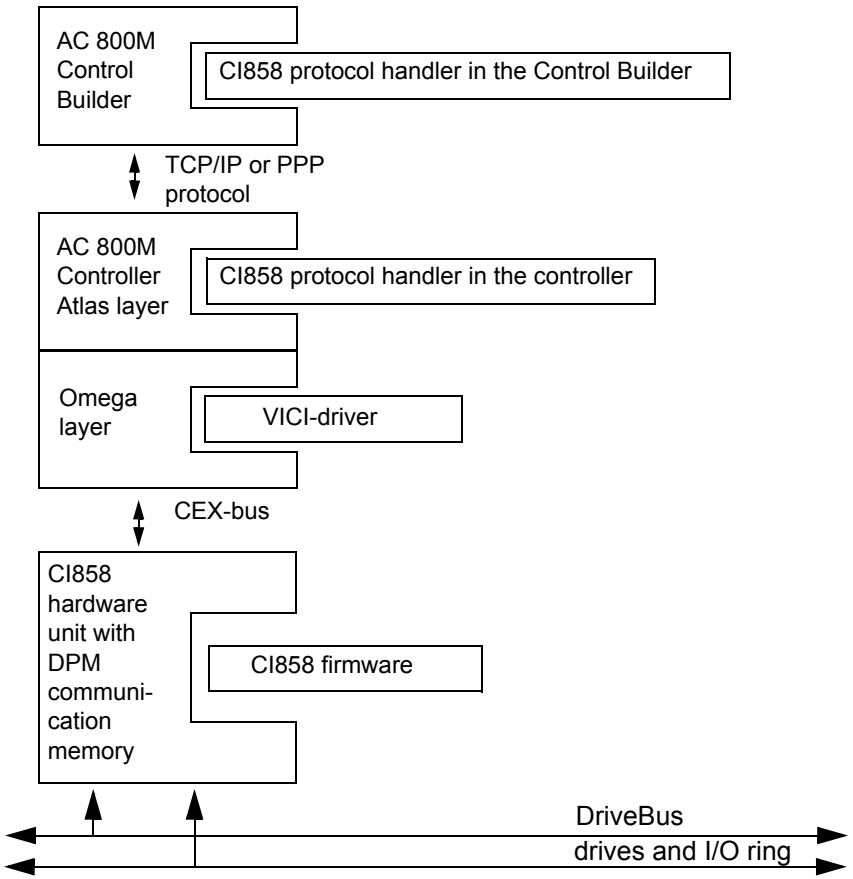
## Introduction

The CI858 software consists of three parts: AC 800M controller software, Control Builder software and CI858 digital signal processor software.

The most important functions of the CI858 software are:

- Cyclic output/input to/from drives.
- Cyclic data to/from I/O units
- Easy configurability of drives and I/O’s to the AC 800M.
- Identification method, self-checking and preventive systems to avoid incorrect configurations.
- Time synchronization of drives to the common calendar time.
- Communication diagnostics for the application

The following picture presents parts of the CI858 software, and how they reside with the AC 800M hardware and software.




ATLAS is a generic framework code for the AC 800M CPU firmware.

OMEGA is a hardware abstraction layer for the AC 800M CPU firmware.

VICI is a very intelligent CI, communications interface. DPM specification for all new intelligent CI units for AC 800M.

# Dataset Communication

The data exchange between the AC 800M, ABB Drives and I/O units via the CI858 consists of dataset pairs, which include input and output datasets. One dataset (DS) consists of three 16-bit words called data words (DW).

Datasets are read from ABB Drives. Therefore datasets need to be defined by setting ABB Drive dataset parameters during the system configuration. See *Section 4, Configuration* on page 33.

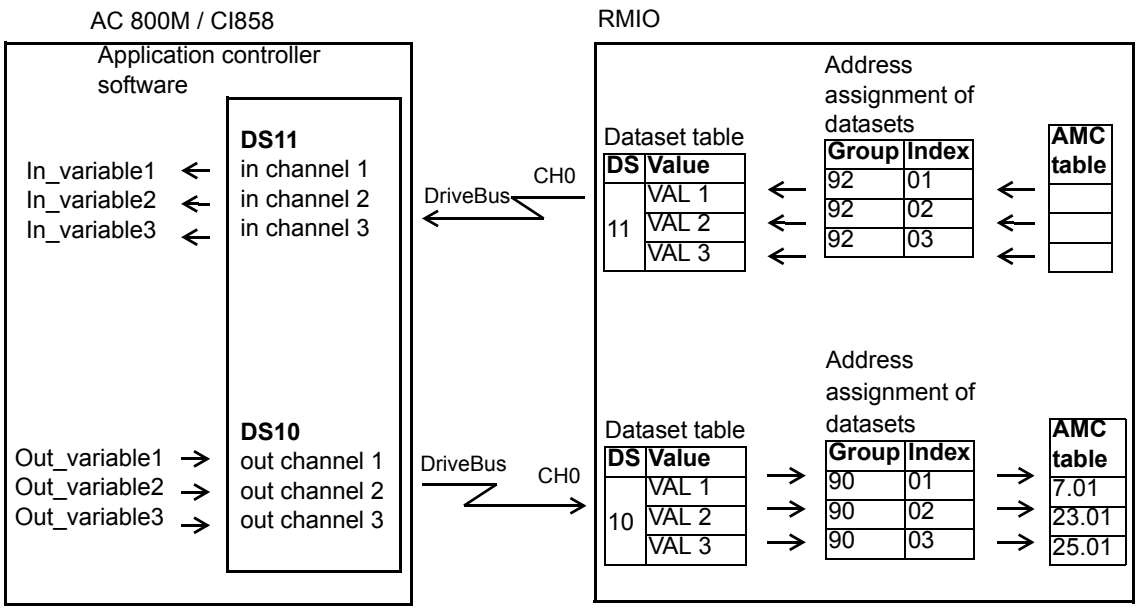



Figure 6. Dataset communication.

## Example of Datasets in the ACS800 Standard Application Program

The ACS800 Standard Application Program 7.x supports the use of four datasets, two in each direction.

The two datasets for controlling the drive are referred to as the Main Reference dataset and the Auxiliary Reference dataset. The sources from which the drive reads the Main and Auxiliary Reference datasets are defined by parameters 90.04 and 90.05 respectively. The contents of the Main Reference dataset are fixed. The contents of the Auxiliary Reference dataset can be selected using parameters 90.01, 90.02 and 90.03.

The two datasets containing actual information on the drive are referred to as the Main Actual Signal dataset and the Auxiliary Actual Signal dataset. The contents of both datasets are partly selectable with the parameters at group 92.

External control is activated by setting parameter 98.02 COMM. MODULE LINK to ADVANT.

Table 1. ACS800 Standard Application Program datasets

| **Data from advant controller to drive** | | |
|---|---|---|
| **Word** | **Contents** | **Selector** |
| **Main Reference dataset DS1** | | |
| 1st Word | Control Word | (Fixed) |
| 2nd Word | Reference 1 | (Fixed) |
| 3rd Word | Reference 2 | (Fixed) |
| **Auxiliary Reference dataset DS3** | | |
| 1st Word | Reference 3 | Par.90.01 |
| 2nd Word | Reference 4 | Par.90.02 |
| 3rd Word | Reference 5 | Par.90.03 |

| **Data from drive to advant controller** | | |
|---|---|---|
| **Word** | **Contents** | **Selector** |
| **Main actual Signal dataset DS2** | | |
| 1st Word | Status Word | (Fixed) |
| 2nd Word | Actual 1 | *Par.92.02 |
| 3rd Word | Actual 2 | Par.92.03 |
| **Aux. Actual Signal dataset Ds4** | | |
| 1st Word | Actual 3 | Par.90.04 |
| 2nd Word | Actual 4 | Par.90.05 |
| 3rd Word | Actual 5 | Par.90.06 |

*Actual 1 is fixed to actual signal 01.02 SPEED (in DTC motor control mode) or 01.03 FREQUENCY (in Scalar mode).

## Example of Datasets in the ACS800 System Application Program

The System Application Program supports the use of datasets 1, 2 or datasets 10...33. Datasets 10...25 are used for data exchange between the AC 800M, ABB Drives and I/O units via the CI858.

Datasets 10...25 are activated by setting parameter 98.02 COMM MODULE to FBA DSET10. Every dataset has a specified read and write task interval in the drive program. The contents of datasets can be selected using parameter Groups 90...93.

Table 2. An example of the System Application datasets.

| Data from advant controller to drive | | | |
|---|---|---|---|
| **DS no.** | **Word** | **Contents** | **Selector** |
| 10 | 1st Word | Main Ctrl Word | Par.90.01 |
| | 2nd Word | Speed Ref | Par.90.02 |
| | 3rd Word | Torque Ref A | Par.90.03 |
| 12 | 1st Word | Reference 3 | Par.90.04 |
| | 2nd Word | Reference 4 | Par.90.05 |
| | 3rd Word | Reference 5 | Par.90.06 |
| 14, 16, 18, 20 | 1st Word | | Par.90.07... 90.18 |
| | 2nd Word | | |
| | 3rd Word | | |
| 22,24 | 1st Word | | Par.91.01 ... 90.06 |
| | 2nd Word | | |
| | 3rd Word | | |

| Data from drive to advant controller | | | |
|---|---|---|---|
| **DS no.** | **Word** | **Contents** | **Selector** |
| 11 | 1st Word | Main status word | Par.92.01 |
| | 2nd Word | Speed measured | Par.92.02 |
| | 3rd Word | Torque Ref B | Par.92.03 |
| 13 | 1st Word | Aux Status Word | Par.92.04 |
| | 2nd Word | Motor Speed | Par.92.05 |
| | 3rd Word | Torque | Par.92.06 |
| 15 | 1st Word | Fault Word 1 | Par.92.07 |
| | 2nd Word | Fault Word 2 | Par.92.08 |
| | 3rd Word | Fault Word 3 | Par.92.09 |
| 17 | 1st Word | Alarm 1 | Par.92.10 |
| | 2nd Word | Alarm 2 | Par.92.11 |
| | 3rd Word | | Par.92.12 |
| 19 | 1st Word | Limit Word 1 | Par.92.12 |
| | 2nd Word | Limit Word 2 | Par.92.13 |
| | 3rd Word | | Par.92.14 |
| 21 | 1st Word | Temperature (of the heatsink) | Par.92.16 |
| | 2nd Word | Motor Meas Temp | Par.92.17 |
| | 3rd Word | | Par.92.18 |
| 23,25 | 1st Word | | Par.93.01... Par 93.06 |
| | 2nd Word | | |
| | 3rd Word | | |

# Section 4 Configuration

## Introduction

To activate communication between AC 800M, CI858, ABB Drives and I/O units the system must be configured with valid parameters:

- Configure the CI858 system with the Control Builder engineering tool.
- Define the system datasets by setting the ABB Drive dataset parameters, for example parameter groups 90...93 for Engineered Drives. See appropriate ABB Drive *Firmware Manual* for dataset and other required parameter settings.

## Control Builder

CI858 system is configured with the Control Builder engineering tool. The configuration includes the following steps:

1. Add units to the hardware tree.
2. Define parameters.
3. Connect variables.
4. Download the project to the controller when all the required steps have been completed.

   **Note:** The CI858 network can be halted during download.

The CI858 and it's units are inserted and defined in the hardware tree with the following hierarchical levels

- CI858
- Drive channel and Special I/O channel
- drives and I/O units

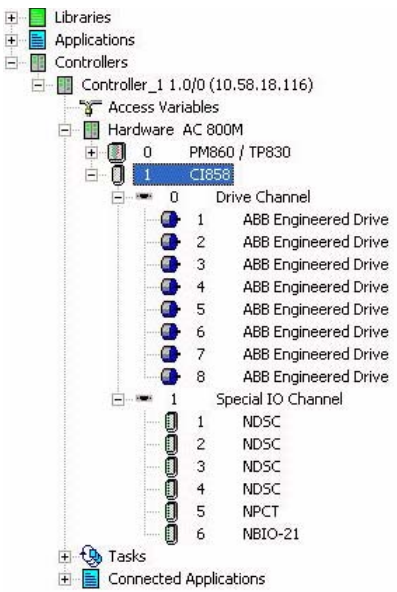



Figure 7. Control Builder's hardware tree in the Project Explorer.

The Control Builder hardware editor contains the following tabs:

**Settings Tab:** Under the Settings tab you define the parameters.

**Connections Tab:** Under the Connections tab you connect the variables to the hardware channels.

**Status Tab:** Under the Status tab you can see the current value and status of each channel.

**Unit Status Tab:** Under the Unit Status tab you can see the current and latched status of the unit.

Figure 8. Control Builder hardware editor tabs.

The Control Builder SFC (The Sequential Function Chart) programming editor is divided into four panes: Declaration pane, Code pane, Structure pane and Message pane.

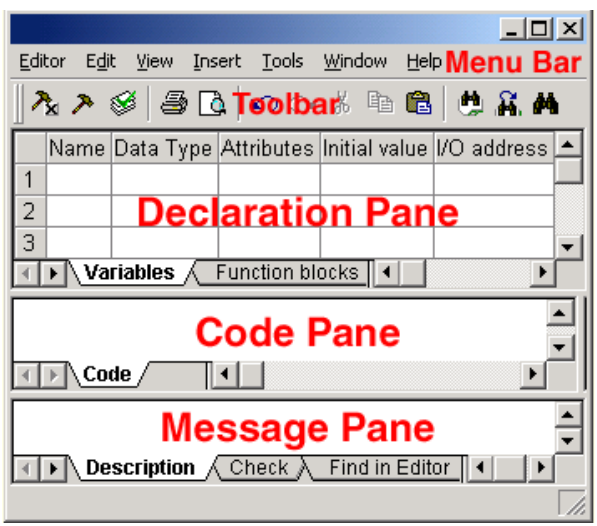



Figure 9. Control Builder programming editor panes.

# CI858 Configuration

## Adding a CI858 Unit

To add a CI858 to the controller proceed as follows in the Project Explorer:

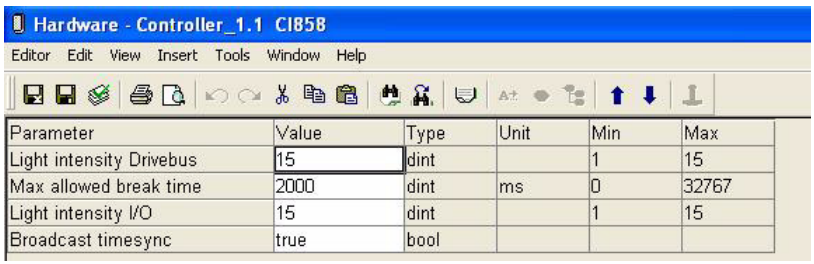
- Right-click the AC 800M hardware object and select New Unit.
- Select a CI858 unit.
- Select the position for the unit and click ok.
- Double-click on the unit and adjust the CI858 parameters in the Settings tab by double-clicking in the value fields. See *CI858 Parameters* on page 36.
- Apply and close the editor (Ctrl + U).

## CI858 Parameters

The following parameters can be specified in the CI858 configuration editor in the Settings tab.

Table 3. DriveBus CI858 parameters.

| Parameter | Selection | Description | Default value |
|---|---|---|---|
| Light intensity DriveBus | 1...15 | With the maximum length of optic fiber cable, use value 15. | 15 |
| Max allowed break time | 0...32767 ms | If there has not been any communication during this time, the connection is closed. The recommended value is 1000-2000 ms. | 2000 |
| Light Intensity I/O | 1...15 | With the maximum length of optic fibre cable, use value 15. | 15 |
| Broadcast timesync | True/False | If the parameter is set to True, the AC 800M clock will synchronize the drive clocks. The resolution is 100 us. | True |




| Parameter | Value | Type | Unit | Min | Max |
|---|---|---|---|---|---|
| Light intensity Drivebus | 15 | dint | | 1 | 15 |
| Max allowed break time | 2000 | dint | ms | 0 | 32767 |
| Light intensity I/O | 15 | dint | | 1 | 15 |
| Broadcast timesync | true | bool | | | |

Figure 10. The CI858 hardware configuration window.

# Drive Configuration

## Adding a Drive

To add a drive to the CI858 proceed as follows in the Project Explorer:

- Right-click the CI858 unit and select New Unit.
- Select Drive Channel
- Right-click the Drive Channel and select New Unit.
- Select the appropriate drive option:
  - ABB Standard Drive with two dataset pairs. (Standard Drive option is used with Standard Application firmware.)
  - ABB Engineered Drive with eight dataset pairs. (Engineered Drive option is used with System Application firmware.)
  - ABB Drive Template (basic) with eight dataset pairs. With ABB Drive Template (extension) the dataset pair number can be extended. With ABB Drive Templates the dataset numbers are user-defined.

**Note:** With ABB Drive products it is always recommended to use ABB Standard Drive or ABB Engineered Drive option.

- Select the position (address of the drive) for the unit and click ok.

  **Note:** Use consecutive numbering between ABB Drive Template (basic) and ABB Drive Template (extension). The position number of the ABB Drive Template (basic) determines the address of the drive associated with the basic and extension templates.
- Double-click on the unit and adjust the drive parameters in the Settings tab by double-clicking in the value fields. See *Drive Parameters* on page 39.
- Connect variables to the channels in the Connections tab. See *Connections* on page 44.
- Apply and close the editor (Ctrl + U).

## Drive Parameters

### ABB Standard Drive and ABB Engineered Drive

The following parameters can be specified in the ABB Standard Drive or in the ABB Engineered Drive configuration editor in the Settings tab.

Table 4. ABB Standard Drive and ABB Engineered Drive parameters.

| Parameter | Selection | Description |
|---|---|---|
| Use BusManager | True/False | If parameter is set to True, the BusManager monitors the connection.<br>The default value is True. |
| Address of Application ID in Drive | 1...9999 | ABB Drive parameter index, which defines the application ID. For example 9910 = parameter 99.10. |
| Configured application ID | 1...65535 | Indicates what the application ID of the drive should be. |
| Configured application ID data type | No Checking / Check Numbers | If Check Numbers is selected, the application checks that the Configured application ID parameter value is correct. |
| Address of drive type in drive | 1...9999 | ABB Drive parameter index, which defines the drive type. For example 401 = parameter 4.01. |
| Drive type | | Defines the drive firmware type. For example AMXG6000 |
| Drive type checking mask | | Defines the allowed drive firmware type.<br>For example AM??????, where '?' is a wild card.<br>The ABB Drive firmware version is compared to this value. |
| Dataset x config status | off/read/ read&write | Defines the dataset access in the Connections tab. |
| Dataset x priority | high priority / normal priority | Defines the dataset priority. It is possible to specify that 1-4 (depending on the drive type) datasets have higher communication priority than the others. |

### ABB Drive Template (basic/extension)

The following parameters can be specified in the ABB Drive Template (basic/extension) configuration editor in the Settings tab.

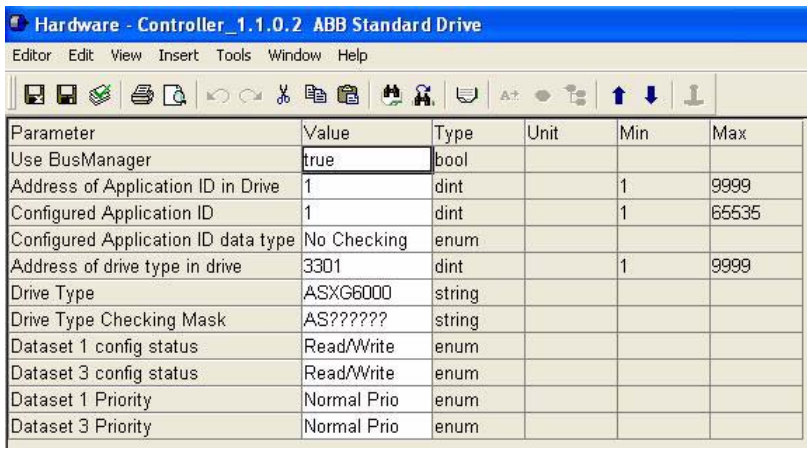
Table 5. ABB Drive Template (basic/extension) parameters.

| Parameter | Selection | Description |
|---|---|---|
| Use BusManager | True/False | If parameter is set to True, the BusManager monitors the connection.<br>The default value is True. |
| Address of Application ID in Drive | 1...9999 | ABB Drive parameter index, which defines the application ID. For example 9910 = parameter 99.10. |
| Configured application ID | 1...65535 | Indicates what the application ID of the drive should be. |
| Configured application ID data type | No Checking / Check Numbers | If Check Numbers is selected, the application checks that the Configured application ID parameter value is correct. |
| Address of drive type in drive | 1...9999 | ABB Drive parameter index, which defines the drive type. For example 401 = parameter 4.01. |
| Drive type | | Defines the drive firmware type. For example AMXG6000 |
| Drive type checking mask | | Defines the allowed drive firmware type.<br>For example AM??????, where '?' is a wild card.<br>The ABB Drive firmware version is compared to this value. |
| Dataset x ID | 0...200 | ABB Drive parameter index, which defines the dataset ID. |
| Dataset x config status | off/read/ read&write | Defines the dataset access in the Connections tab. |
| Dataset x priority | high priority / normal priority | Defines the dataset priority. It is possible to specify that 1-4 (depending on the drive type) datasets have higher communication priority than the others. |

**BusManager**

The BusManager is used for reading and checking the actual drive configuration. It is also used for recovery of broken communication links. Normal communication to the broken link is suspended until the link is reported to be recovered.

If the BusManager function is disabled, the communication is assumed to be cyclic and continuous. A missing drive causes extra communication, because the messages towards it are repeated continuously.




| Parameter | Value | Type | Unit | Min | Max |
|---|---|---|---|---|---|
| Use BusManager | true | bool | | | |
| Address of Application ID in Drive | 1 | dint | | 1 | 9999 |
| Configured Application ID | 1 | dint | | 1 | 65535 |
| Configured Application ID data type | No Checking | enum | | | |
| Address of drive type in drive | 3301 | dint | | 1 | 9999 |
| Drive Type | ASXG6000 | string | | | |
| Drive Type Checking Mask | AS?????? | string | | | |
| Dataset 1 config status | Read/Write | enum | | | |
| Dataset 3 config status | Read/Write | enum | | | |
| Dataset 1 Priority | Normal Prio | enum | | | |
| Dataset 3 Priority | Normal Prio | enum | | | |

Figure 11. ABB Standard Drive parameter window.

# Special I/O Configuration

## Adding a Special I/O Unit

Before Special I/O Channel can be selected, the Special I/O Channel Hardware definition file has to be imported into the project.

To add a Special I/O unit to the CI858 proceed as follows in the Project Explorer:

- Right-click the CI858 unit and select New Unit.
- Select Special I/O channel.
- Right-click the Special I/O channel and select New Unit.
- Select the appropriate I/O unit.
- Select the position (address of the I/O unit) for the unit and click ok.
- Double-click on the unit and adjust the Special I/O parameters in the Settings tab by double-clicking in the value fields. See *Special I/O Parameters* on page 43.
- Connect variables to the channels in the Connections tab. See *Special I/O Function Blocks* on page 46 and *Connections* on page 44.
- Apply and close the editor (Ctrl + U).

**Note:** ID number of the I/O unit in the hardware tree has to match the device number configured with switch S1 on the unit.

## Special I/O Parameters

The following parameters can be specified in the Special I/O configuration editor in the Settings tab for the available Special I/O units: Special I/O Template, NBIO-21, NBIO-31, NWIO, NAIO, NDIO, NTAC, NPCT, NDSC, NCTI.

Table 6. Special I/O parameters

| Parameter | Selection | Description |
|---|---|---|
| Configured device type | | Select the appropriate I/O type. |
| Initialization style for device | | Parameter is set automatically according to the configured device type. In the Special I/O Template parameter is user-defined<br>Defines the ASIC type. Set parameter to 0 or 1. Select 1 if the ASIC is of the IOCC type. |
| MAFR Code | | Major function code. Parameter is set automatically according to the configured device type. In the Special I/O Template parameter is user-defined.<br>Defines whether the dataset is I/O or drive related. Set parameter to 0...255. |
| Overlayed datasets | | Parameter is set automatically according to the configured device type. In the Special I/O Template parameter is user-defined.<br>Set parameter to True or False. Select True if the returned dataset has the same dataset ID. |
| Dataset x ID | 0...200 | ABB Drive parameter index, which defines the Special I/O Template dataset ID. |
| Dataset x config status(1 | off/read/ read&write | Defines the dataset access in the Connections tab. |
| Dataset x priority | high priority / normal priority | Defines the dataset priority. It is possible to specify that 1-4 (depending on the drive type) datasets have higher communication priority than the others. |
| 1) Configuration status for datasets for reading digital inputs are marked with R (read) and dataset for reading and writing digital outputs are marked with R/W (read and write). | | |

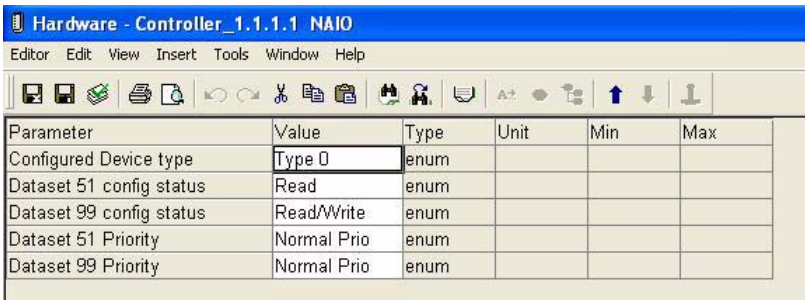

Hardware - Controller_1.1.1.1 NAIO
Editor Edit View Insert Tools Window Help
Parameter Value Type Unit Min Max
Configured Device type Type 0 enum
Dataset 51 config status Read enum
Dataset 99 config status Read/Write enum
Dataset 51 Priority Normal Prio enum
Dataset 99 Priority Normal Prio enum

Figure 12. NAIO parameter window.

**Special I/O Template**

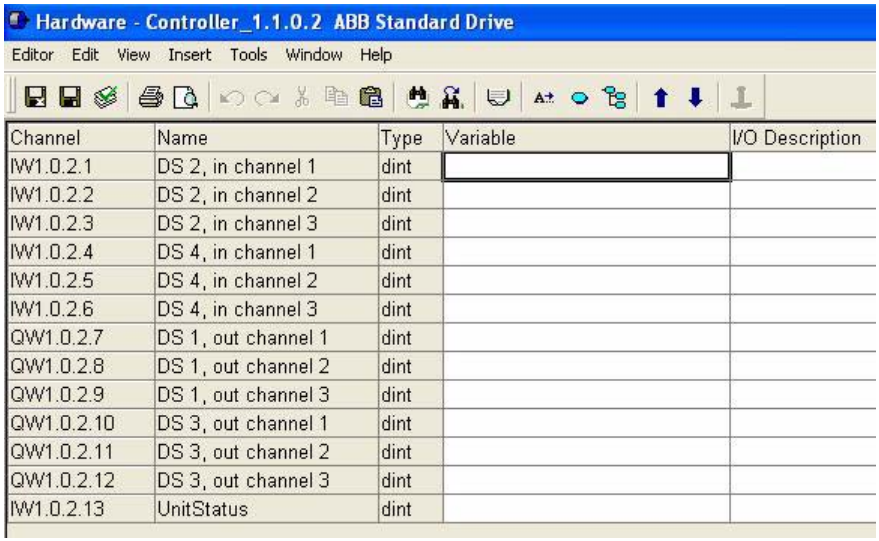
With Special I/O Template, the dataset numbers are user-defined.

# Connections

Variables are connected to the input and output dataset channels in the connections window shown below.

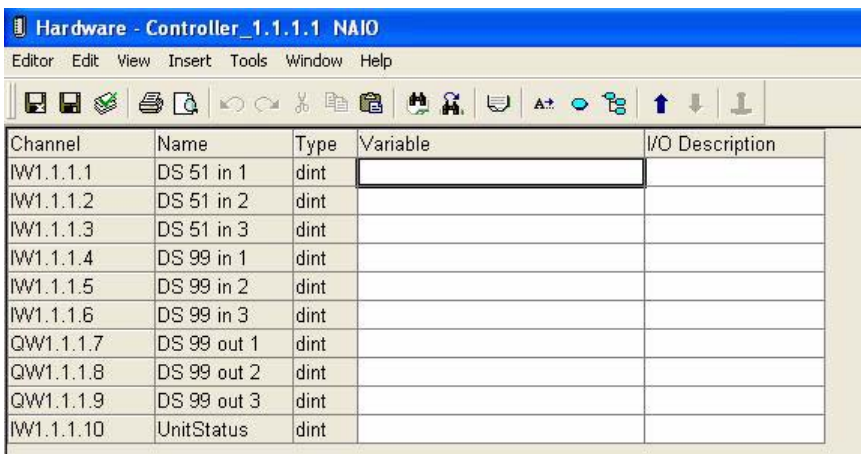
**Note:** All connected variables must be in the same task: All connected, accessed variables will be copied to the same task. All connected, unaccessed variables will be copied to a background task.

**Note:** When dataset access is defined as Read&Write, variables of at least one input channel and one output channel must be connected. With Read only datasets variables of all three input channels must be connected.




| Channel | Name | Type | Variable | I/O Description |
|---|---|---|---|---|
| IW1.0.2.1 | DS 2, in channel 1 | dint | | |
| IW1.0.2.2 | DS 2, in channel 2 | dint | | |
| IW1.0.2.3 | DS 2, in channel 3 | dint | | |
| IW1.0.2.4 | DS 4, in channel 1 | dint | | |
| IW1.0.2.5 | DS 4, in channel 2 | dint | | |
| IW1.0.2.6 | DS 4, in channel 3 | dint | | |
| QW1.0.2.7 | DS 1, out channel 1 | dint | | |
| QW1.0.2.8 | DS 1, out channel 2 | dint | | |
| QW1.0.2.9 | DS 1, out channel 3 | dint | | |
| QW1.0.2.10 | DS 3, out channel 1 | dint | | |
| QW1.0.2.11 | DS 3, out channel 2 | dint | | |
| QW1.0.2.12 | DS 3, out channel 3 | dint | | |
| IW1.0.2.13 | UnitStatus | dint | | |

Figure 13. ABB Standard Drive connections window.




| Channel | Name | Type | Variable | I/O Description |
|---|---|---|---|---|
| IW1.1.1.1 | DS 51 in 1 | dint | | |
| IW1.1.1.2 | DS 51 in 2 | dint | | |
| IW1.1.1.3 | DS 51 in 3 | dint | | |
| IW1.1.1.4 | DS 99 in 1 | dint | | |
| IW1.1.1.5 | DS 99 in 2 | dint | | |
| IW1.1.1.6 | DS 99 in 3 | dint | | |
| QW1.1.1.7 | DS 99 out 1 | dint | | |
| QW1.1.1.8 | DS 99 out 2 | dint | | |
| QW1.1.1.9 | DS 99 out 3 | dint | | |
| IW1.1.1.10 | UnitStatus | dint | | |

Figure 14. NAIO connection window.

# Special I/O Function Blocks

This chapter introduces the Special I/O units supported by the CI858 and gives instructions on how to configure the input and output channels of the I/O units in the application program.

I/O units can be controlled with function blocks. All connected I/O units require their own function block, which is selected according to the I/O unit type. A function block has a defined set of input and output variables and variables for internal storage. It also has an algorithm to operate these variables. When a function block is executed, it evaluates all its variables and computes new values for the output and internal variables.

Special I/O function blocks are stored in the Special I/O library delivered with the product. By connecting the library to a project in the Control Builder, the function blocks become available to the application.

**Note:** Before using function blocks, appropriate library must be inserted into the project and the I/O unit must be inserted into the CI858 hardware tree.

#### How to Insert a Library File

1. Select from the Control Builder menu Insert File Into Project - Library.
2. The Insert Library dialog box is displayed. Select a library from the list in the dialog box or type the name of the library in the File name field (extension *.lbr).
3. Click the Open button to insert the library.
4. The inserted library is displayed in the Project Explorer as a subfolder to the Library folder.

#### How to Insert a Special I/O Unit

See *Adding a Special I/O Unit* on page 42.

**Note:** More information on how to configure and use Special I/O is described in the Special I/O documentation. This documentation, together with the Special I/O library and Special I/O Hardware file, is delivered with the Special I/O product.

## Activating a Function Block

A Function block is activated in the programming editor.

1. Expand the root Application(s) folder and select the Programs folder. Double-click the appropriate program to enter the programming editor.

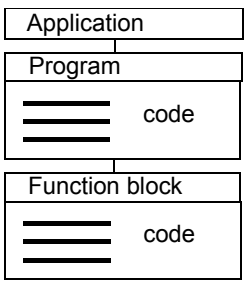



Figure 15. Hierarchical structure of the application.

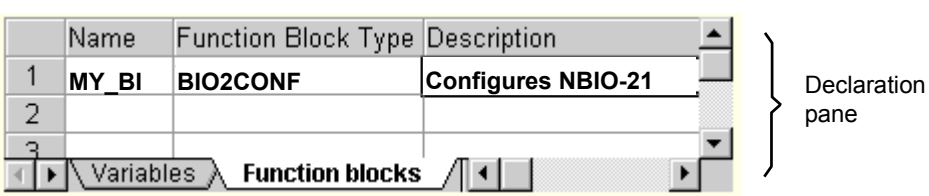
2. Select Function blocks tab in the Declaration pane and fill in the following items:

   **Name:** User-defined name of the function block.

   **Function Block Type:** Name of the function block used in the library file as defined in this manual. See the following function block descriptions.

   Task Connection: This section needs to be filled only if distributed execution is applied to the application program.

   Description: User-defined information about the function block.



3. Type the user-defined name of the function block followed by a left parenthesis in the Code pane. A connection editor is automatically displayed to assist you in specifying the input/output parameters. (You cannot edit the parameter list itself, you can only connect already defined parameters.)

**Note:** Each I/O unit in the hardware tree must have their own function block. I.e. if there are tree NTAC-02 units, tree TAC_SP function blocks need to called.

## NBIO-21

The Basic I/O unit NBIO-21 has 3 digital inputs, 2 analogue inputs, 2 digital outputs and 2 analogue outputs.

There are five library function blocks for handling the NBIO-21 unit: BIO2CONF, BIO2DI, BIO2DO, BIO2AO and BIO2AI.

#### BIO2CONF

Function block BIO2CONF is used to configure the NBIO-21 unit. BIO2CONF selects the hardware gain of the analogue inputs and provides automatic calibration for the A/D converter.

BIO2CONF should be called if function block BIO2AI is in use, and if the default configuration of the unit is not sufficient. To reduce system load, BIO2CONF is recommended to be executed at a longer time cycle than BIO2AI, for example > 1000 ms.

The actual selection of the I/O unit is made in Control Builder's hardware editor by connecting respective channels of the NBIO-21 unit to input and output signals 4...12 (conf_inw1... UnitStatus). See *Table 7* on page 49 and *Connections* on page 44.

**Note:** Cyclic DDCS messages from BIO2CONF cannot be disabled.

Table 7. Input and output signals of the BIO2CONF.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | CONV_AI1 | in | bool | Hardware gain for channel 1. See table 8. |
| 2 | CONV_AI2 | in | bool | Hardware gain for channel 2. See table 8. |
| 3 | ERR | out | bool | Communication status. 1 = Error<br>See *Output Error* on page 82. |
| 4 | conf_inw1 | in | dint | First input word of DS 101. |
| 5 | conf_outw1 | out | dint | First output word of DS 101. |
| 6 | conf_outw2 | out | dint | Second output word of DS 101. |
| 7 | conf_outw3 | out | dint | Third output word of DS 101. |
| 8 | calib_inw | in | dint | First input word of DS 32. |
| 9 | calib_outw1 | out | dint | First output word of DS 32. |
| 10 | calib_outw2 | out | dint | Second output word of DS 32. |
| 11 | calib_outw3 | out | dint | Third output word of DS 32. |
| 12 | UnitStatus | in | dint | Unit Status |

Table 8. Hardware gain of the analogue inputs.

| Type | Values | Channel AI1 | Channel AI2 |
|---|---|---|---|
| Voltage | -10...10 V | CONV_AI1 = 0 | CONV_AI2 = 0 |
| Voltage | -2...2 V | CONV_AI1 = 1 | CONV_AI2 = 1 |
| Current | -20...20 mA | CONV_AI1 = 1 | CONV_AI2 = 1 |

### BIO2DI

Function block BIO2DI is used for reading digital input signals from the NBIO-21 unit.

The actual selection of the I/O unit is made in Control Builder's hardware editor by connecting respective channels of the NBIO-21 unit to input signals 5...8 (inword1... UnitStatus). See *Table 9* on page 50 and *Connections* on page 44.

Table 9. Input and output signals of the BIO2DI.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | DI1 | out | bool | Output for digital input DI1. |
| 2 | DI2 | out | bool | Output for digital input DI2. |
| 3 | DI3 | out | bool | Output for digital input DI3. |
| 4 | ERR | out | bool | Communication status. 1 = error<br>See *Output Error* on page 82. |
| 5 | Inword1 | in | dint | First input word of DS 100 (R). |
| 6 | Inword2 | in | dint | Second input word of DS 100 (R). |
| 7 | Inword3 | in | dint | Third input word of DS 100 (R). |
| 8 | UnitStatus | in | dint | Unit status |

### BIO2DO

Function block BIO2DO is used for writing digital output signals to the NBIO-21 unit.

The actual selection of the I/O unit is made in Control Builder's hardware editor by connecting respective channels of the NBIO-21 unit to input and output signals 4...8 (outword1... UnitStatus). See *Table 10* on page 51 and *Connections* on page 44.

Table 10. Input and output signals of the BIO2DO.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | DO1 | in | bool | Input for digital output DO1 |
| 2 | DO2 | in | bool | Input for digital output DO2 |
| 3 | ERR | out | bool | Communication status. 1 = Error<br>See *Output Error* on page 82. |
| 4 | outword1 | out | dint | First output word of DS 100 (R/W). |
| 5 | outword2 | out | dint | Second output word of DS 100 (R/W). |
| 6 | outword3 | out | dint | Third output word of DS 100 (R/W). |
| 7 | inword1 | in | dint | First input word (= dummy) of DS 100 (R/W). |
| 8 | UnitStatus | in | dint | Unit status. |

**BIO2AO**

Function block BIO2AO is used for writing analogue output signals (voltage/current) to the NBIO-21 unit.

The actual selection of the I/O unit is made in Control Builder's hardware editor by connecting respective channels of the NBIO-21 unit to input and output signals 4...8 (dummy_in... UnitStatus). See *Table 11* on page 52 and *Connections* on page 44.

Table 11. Input and output signals of the BIO2AO.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | AO1 | in | dint | Input for analogue output AO1. |
| 2 | AO2 | in | dint | Input for analogue output AO2. |
| 3 | ERR | out | bool | Communication status. 1 = Error<br>See *Output Error* on page 82. |
| 4 | dummy_in | in | dint | First input word (=dummy) of DS 99. |
| 5 | outword1 | out | dint | First output word of DS 99. |
| 6 | outword2 | out | dint | Second output word of DS 99. |
| 7 | outword3 | out | dint | Third output word of DS 99. |
| 8 | UnitStatus | in | dint | Unit status |

BIO2AO does not scale the analogue output signals. Scaling between application software and analogue output AO1/AO2 requires extra application blocks.

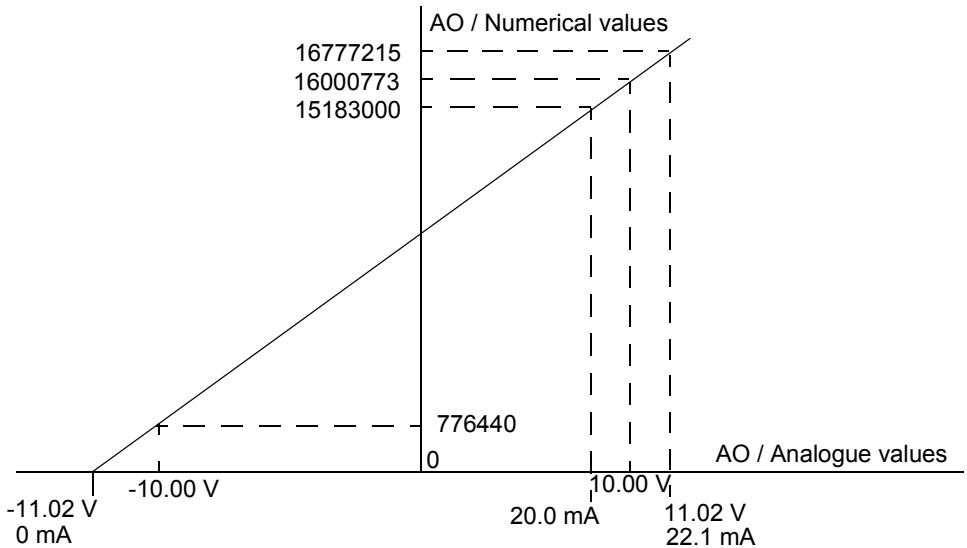
*Scaling example:*

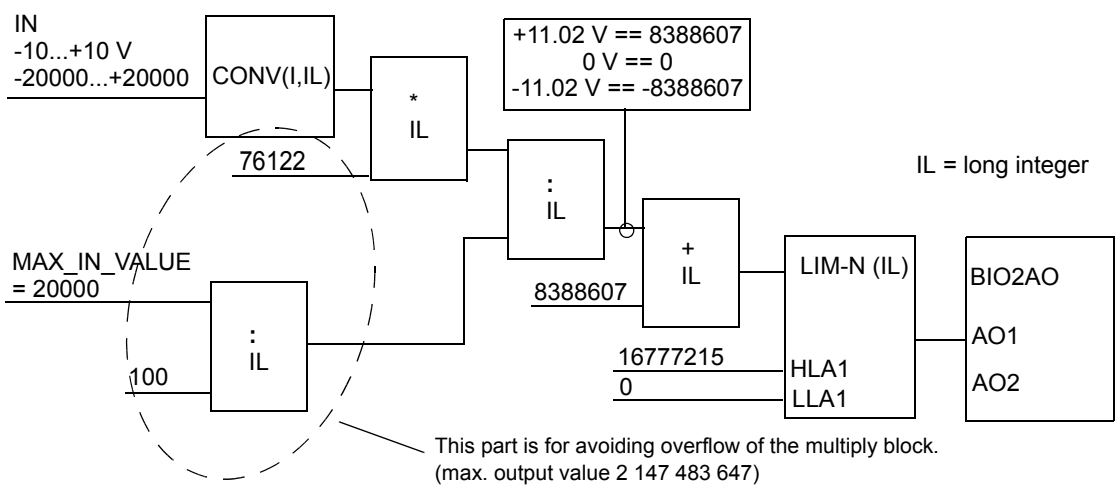
Analogue output value -10 V…+10 V corresponds to integer (I) input value -20000…+20000 in the application software.

$AO1 = (IN * VALUE_{10V} / MAX\_IN\_VALUE) + 8388607$
$MAX\_IN\_VALUE = 20000$
$VALUE_{10V} = (10.00 / 11.02) * 8388607 = 7612166$

Table 12. Analogue output scaling.

| Channel | Type | Nominal value | Scaling for nominal value | Full scaling range |
|---|---|---|---|---|
| AO1, AO2 | Voltage | -10.00...+10.00 V | 776440…16000773 | -11.02…+11.02 V<br>0...16777215 |
| AO1, AO2 | Current | 0...20.0 mA | 0…15183000 | 0…22.10 mA<br>0...16777215 |




Voltage scaling
+10.00 V == (10.00 V / 11.02 V) * 8388607 = 7612166
-10.00 V == (10.00 V / 11.02 V) * -8388607 = -7612166

**BIO2AI**

Function block BIO2AI is used for reading analogue input signals (voltage/current) from the NBIO-21 unit.

The actual selection of the I/O unit is made in Control Builder's hardware editor by connecting respective channels of the NBIO-21 unit to input signals 4...7 (inword1... UnitStatus). See *Table 13* on page 54 and *Connections* on page 44.

Voltage/current measurement is selected using a 2-pole DIP switch S2 on the NBIO-21 unit.

Table 13. Input and output signals of the BIO2AI.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | AI1 | out | dint | Output for analogue input AI1. |
| 2 | AI2 | out | dint | Output for analogue input AI2. |
| 3 | ERR | out | bool | Communication status. 1 = Error<br>See *Output Error* on page 82. |
| 4 | inword1 | in | dint | First input word of DS 54. |
| 5 | inword2 | in | dint | Second input word of DS 54. |
| 6 | inword3 | in | dint | Third input word of DS 54. |
| 7 | UnitStatus | in | dint | Unit status |

BIO2AI does not scale the analogue input signals. Scaling between application software and analogue input requires extra application blocks.

*Scaling example 1:*

Analogue input value -10 V …+10 V corresponds to integer (I) value -20000…+20000 in the application software.

$AI1_{scaled}$ = (AI1 * MAX_OUT_VALUE) / $VALUE_{10V}$
MAX_OUT_VALUE = 20000
$VALUE_{10V}$ = (10.00 / 11.25) * 8388607 = 7456540

*Scaling example 2:*

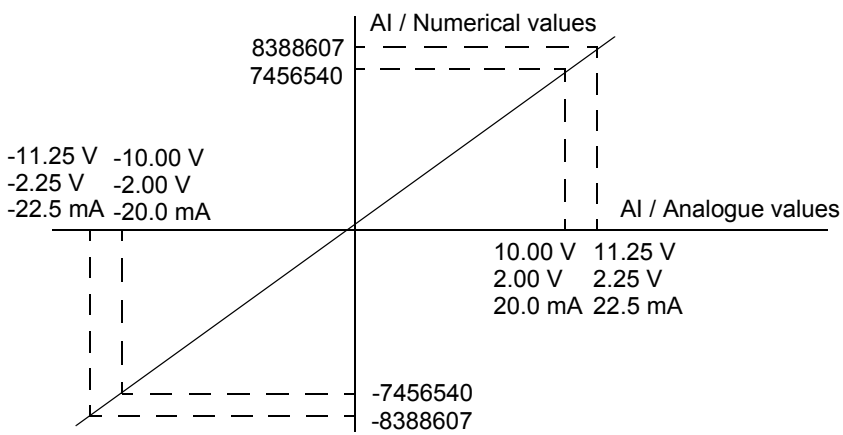
Analogue input value 4 mA …20 mA corresponds to integer (I) value 0…+20000 in the application software.

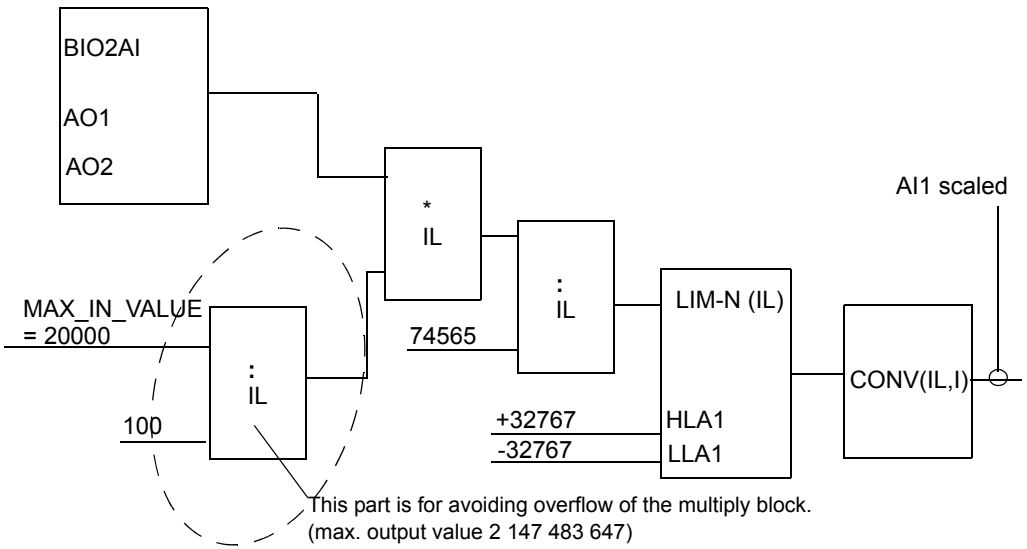
$AI1_{scaled}$ = ((AI1 - $VALUE_{4mA}$) * MAX_OUT_VALUE) / $VALUE_{16mA}$
MAX_OUT_VALUE = 20000
$VALUE_{4mA}$ = (4 / 22.5) * 8388607 = 1491308
$VALUE_{16mA}$ = (16 / 22.5) * 8388607 = 5965231

Scaled output AI1 value must be limited between 0...32767.

Table 14. Analogue input scaling.

| Channel | Type | Nominal value | Scaling for nominal value | Full scale range |
|---|---|---|---|---|
| AI1, AI2 | Voltage | -10...10 V | -7456540…7456540 | -11.25...11.25 V, +/-8388607 |
| AI1, AI2 | Voltage | -2...2 V | -7456540…7456540 | -2.25...2.25 V, +/-8388607 |
| AI1, AI2 | Current | -20...20 mA | -7456540…7456540 | -22.5...22.5 mA, +/-8388607 |






Voltage scaling
+10.00 V == (10.00 V / 11.25 V) * 8388607 = 7456540
-10.00 V == (10.00 V / 11.25 V) * -8388607 = -7456540

## NBIO-31

The Basic I/O unit NBIO-31 has 4 digital inputs and 3 digital outputs.

There are two library function blocks for handling the NBIO-31 unit: BIO3DI and BIO3DO.

### BIO3DI

Function block BIO3DI is used for reading digital input signals from the NBIO-31 unit.

The actual selection of the I/O unit is made in the Control Builder's hardware editor by connecting respective channels of the NBIO-31 unit to input signals 6...9 (inword1... UnitStatus). See *Table 15* on page 57 and *Connections* on page 44.

Table 15. Input and output signals of the BIO3DI.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | DI1 | out | bool | Output for Digital Input DI1. |
| 2 | DI2 | out | bool | Output for Digital Input DI2. |
| 3 | DI3 | out | bool | Output for Digital Input DI3. |
| 4 | DI4 | out | bool | Output for Digital Input DI4. |
| 5 | ERR | out | bool | Communication Status. 1 = Error<br>See *Output Error* on page 82. |
| 6 | inword1 | in | dint | First input word of DS 100 (R). |
| 7 | inword2 | in | dint | Second input word of DS 100 (R). |
| 8 | inword3 | in | dint | Third input word of DS 100 (R). |
| 9 | UnitStatus | in | dint | Unit Status |

**BIO3DO**

Function block BIO3DO is used for writing digital output signals to the NBIO-31 unit.

The actual selection of the I/O unit is made in the Control Builder's hardware editor by connecting respective channels of the NBIO-31 unit to input and output signals 5...9 (outword1... UnitStatus). See *Table 16* on page 58 and *Connections* on page 44.

Table 16. Input and output signals of the BIO3DO.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | DO1 | in | bool | Input for digital output DO1. |
| 2 | DO2 | in | bool | Input for digital output DO2. |
| 3 | DO3 | in | bool | Input for digital output DO3. |
| 4 | ERR | out | bool | Communication status. 1 = Error<br>See *Output Error* on page 82. |
| 5 | outword1 | out | dint | First output word of DS 100 (R/W). |
| 6 | outword2 | out | dint | Second output word of DS 100 (R/W). |
| 7 | outword3 | out | dint | Third output word of DS 100 (R/W). |
| 8 | inword1 | in | dint | First input word (=dummy) of DS 100 (R/W). |
| 9 | UnitStatus | in | dint | Unit status |

## NDIO

The NDIO unit has two digital inputs and two relay outputs. For more information see *NTAC-0x/NDIO-0x/NAIO-0x Installation and Start-up Guide [3AFY 58919730 (English)]*.

There are two library function blocks for handling NDIO units: NDIO1DI and NDIO1DO.

### NDIO1DI

Function block NDIO1DI is used for reading digital input signals from the NDIO unit.

The actual selection of the I/O unit is made in the Control Builder's hardware editor by connecting respective channels of the NDIO unit to input signals 4...7 (inword1... UnitStatus). See *Table 17* on page 59 and *Connections* on page 44.

Table 17. Input and output signals of the NDIO1DI.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | DI1 | out | bool | Output for digital input DI1. |
| 2 | DI2 | out | bool | Output for digital input DI2. |
| 3 | ERR | out | bool | Communication status. 1 = Error<br>See *Output Error* on page 82. |
| 4 | inword1 | in | dint | First input word of DS 100 (R). |
| 5 | inword2 | in | dint | Second input word of DS 100 (R). |
| 6 | inword3 | in | dint | Third input word of DS 100 (R). |
| 7 | UnitStatus | in | dint | Unit status |

**NDIO1DO**

Function block NDIO1DO is used for controlling digital outputs of the NDIO.

The actual selection of the I/O unit is made in the Control Builder's hardware editor by connecting respective channels of the NDIO unit to input and output signals 4...8 (outword1... UnitStatus). See *Table 18* on page 60 and *Connections* on page 44.

Table 18. Input and output signals of the NDIO1DO.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | DO1 | in | bool | Input for digital output DO1. |
| 2 | DO2 | in | bool | Input for digital output DO2. |
| 3 | ERR | out | bool | Communication status. 1 = Error<br>See *Output Error* on page 82. |
| 4 | outword1 | out | dint | First output word of DS 100 (R/W). |
| 5 | outword2 | out | dint | Second output word of DS 100 (R/W). |
| 6 | outword3 | out | dint | Third output word of DS 100 (R/W). |
| 7 | inword1_dummy | in | dint | First input word (=dummy) of DS 100 (R/W). |
| 8 | UnitStatus | in | dint | Unit status |

## NAIO

The NAIO unit has two current or voltage inputs and two current outputs. For more information see *NTAC-0x/NDIO-0x/NAIO-0x Installation and Start-up Guide [3AFY 58919730 (English)].*

There are two library function blocks for handling NAIO units: NAIO2AI and NAIO2AO.

### NAIO2AO

Function block NAIO2AO is used for writing analogue output signals (current) to the NAIO unit.

The actual selection of the I/O unit is made in the Control Builder’s hardware editor by connecting respective channels of the NAIO unit to input and output signals 4...8 (outword1... UnitStatus). See *Table 19* on page 61 and *Connections* on page 44.

Table 19. Input and output signals of the NAIO2AO.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | AO1 | in | dint | Input for analogue output AO1 |
| 2 | AO2 | in | dint | Input for analogue output AO2. |
| 3 | ERR | out | bool | Communication status. 1 = Error<br>See *Output Error* on page 82. |
| 4 | outword1 | out | dint | First output word of DS 99. |
| 5 | outword2 | out | dint | Second output word of DS 99. |
| 6 | outword3 | out | dint | Third output word of DS 99. |
| 7 | inword1_dummy | in | dint | First input word (=dummy) of DS 99. |
| 8 | UnitStatus | in | dint | Unit status |

NAIO2AO does not scale the analogue output signals. Scaling between application software and analogue output AO1/AO2 requires extra application blocks.

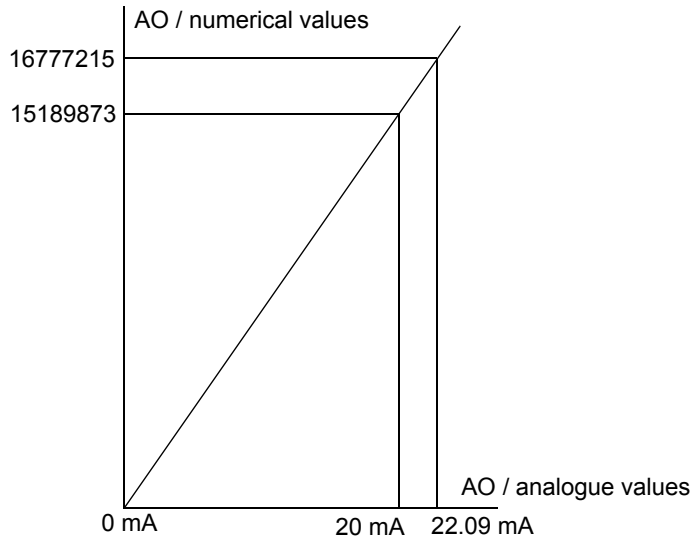
*Scaling example:*

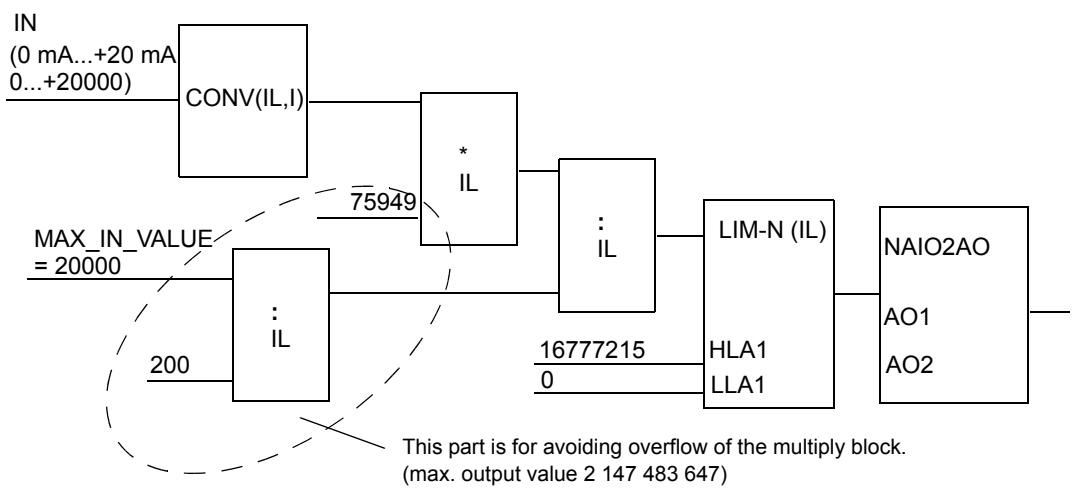
An integer (I) input value 0…+20000 in the application software corresponds to analogue output value 0…+20 mA.

$AO1 = (IN * VALUE_{20mA}) / MAX\_IN\_VALUE$
$MAX\_IN\_VALUE = 20000$
$VALUE_{20mA} = (20.00 / 22.09) * 16777215 = 15189873$

Table 20. Analogue output scaling

| Channel | Type | Nominal value | Scaling for nominal value | Full scaling range |
|---|---|---|---|---|
| AO1, AO2 | Current | 0...20 mA | 0…15189873 | 0…22.09 mA, 0...16777215 |




Current scaling
+20 mA == (20 mA / 22.09 mA) * 16777215 = 15189873
0 mA == 0

### NAIO2AI

Function block NAIO2AI is used for reading analogue input signals (voltage/current) from the NAIO unit.

The actual selection of the I/O unit is made in the Control Builder's hardware editor by connecting respective channels of the NAIO unit to input signals 4...7 (inword1... UnitStatus). See *Table 21* on page 63 and *Connections* on page 44.

Table 21. Input and output signals of the NAIO2AI.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | AI1 | out | dint | Output for digital input DI1. |
| 2 | AI2 | out | dint | Output for digital input DI2. |
| 3 | ERR | out | bool | Communication status. 1 = Error<br>See *Output Error* on page 82. |
| 4 | inword1 | in | dint | First input word of DS 51. |
| 5 | inword2 | in | dint | Second input word of DS 51. |
| 6 | inword3 | in | dint | Third input word of DS 51. |
| 8 | UnitStatus | in | dint | Unit status |

The voltage/current measurement is selected using DIP switches on the NAIO unit.

NAIO2AI does not scale the analogue input signals. Scaling between application software and analogue input requires extra application blocks.

*Scaling example:*

Analogue input value -10…+10 V corresponds to integer (I) value -20000…+20000 in the application software.

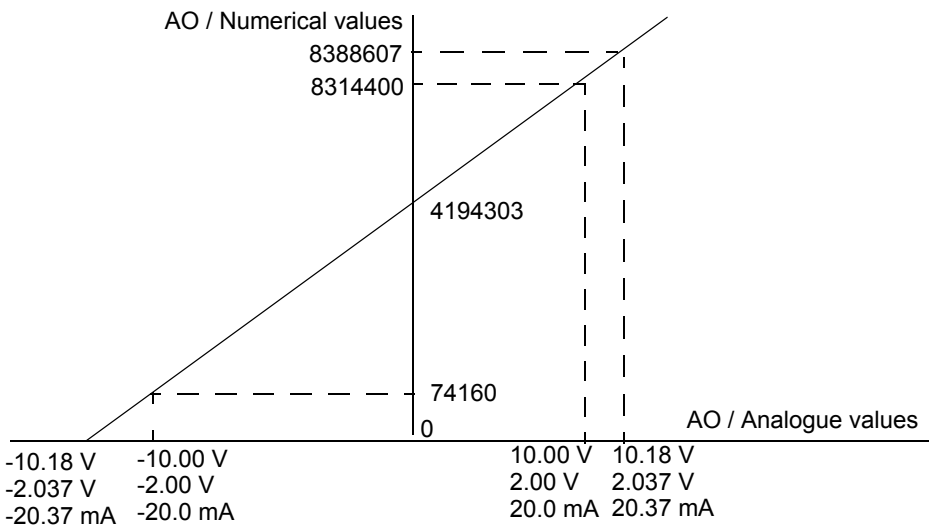
$AI1_{scaled}$ = ((AI1 - VALUEOFFSET_0V) * MAX_OUT_VALUE) / $VALUE_{10V}$
MAX_OUT_VALUE = 20000
VALUEOFFSET_0V = 8388607 / 2 = 4194303
$VALUE_{10V}$ = (10.00 / 10.18) * 8388607 / 2 = 4120140

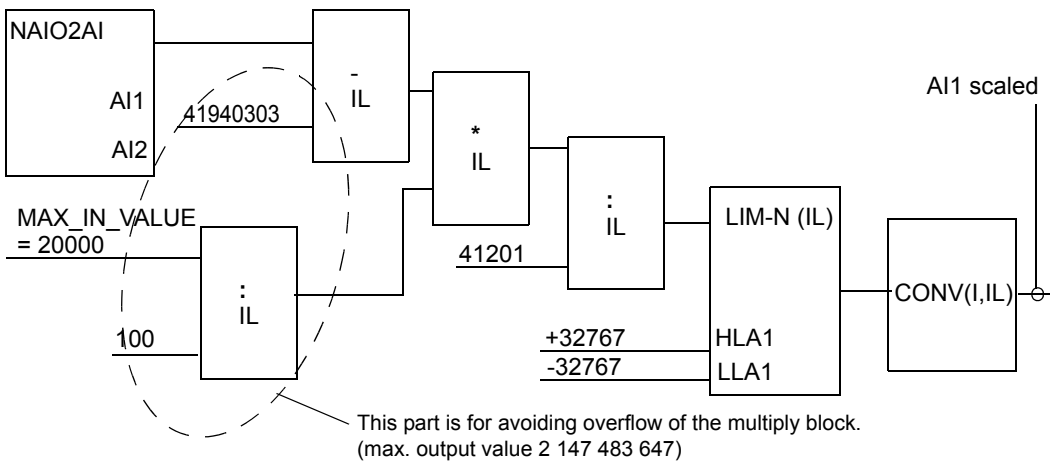
Table 22. Unipolar Analogue Input Scaling.

| Channel | Type | Nominal value | Scaling for nominal value | Full scaling range |
|---|---|---|---|---|
| AI1, AI2 | Voltage | 0...10 V | 0…8240282 | 0...10.18 V, 0…8388607 |
| AI1, AI2 | Voltage | 0...2 V | 0...8240282 | 0...2.037 V, 0…8388607 |
| AI1, AI2 | Current | 0...20 mA | 0...8240282 | 0…20.37mA, 0…8388607 |

Table 23. Bipolar Analogue Input Scaling.

| Channel | Type | Nominal value | Scaling for nominal value | Full scaling range |
|---|---|---|---|---|
| AI1, AI2 | Voltage | -10...10 V | 74160…8314400 | -10.18...10.18 V, 0…8388607 |
| AI1, AI2 | Voltage | -2...2 V | 74160...8314400 | -2.037...2.037 V, 0…8388607 |
| AI1, AI2 | Current | -20...20 mA | 74160...8314400 | -20.37…20.37mA, 0…8388607 |






## NPCT

NPCT-01 Pulse Counter / Timer Unit is an I/O option board used with CI858. Its main areas of use are speed and position measurement and applications which require programmable, hardware based counter functions such as fast control of digital outputs as a function of position or time and holding registers for capturing events. For more information see *NPCT-01 Pulse Counter/Timer Unit description [3AFY64362828(English)].*

There are five library function blocks for handling the NPCT-01 board: PCT_CONF, SPEEDPO1, SPEED_R, POS_IL1 and CONV24.

### PCT_CONF

Function block PCT_CONF is used to change the default configuration of the NPCT-01 board.

The actual selection of the I/O unit is made in the Control Builder's hardware editor by connecting respective channels of the NPCT board to input and output signals 12...20 (outword2_ds3... UnitStatus). See *Table 24* on page 66 and *Connections* on page 44.

Board specific configurations are sent to NPCT board via dataset 5 by using Outword3_ds5. Dataset 3 (encoder channel 1) and dataset 4 (encoder channel 2) are used for channel specific configurations.

Table 24. Input and output signals of the PCT_CONF. (DS= dataset, DW = data word)

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | WD_TO | in | bool | Selection of WatchDog time out. 0 = 100 ms, 1 = 1 s (default) |
| 2 | WD_ENA | in | bool | Enables WatchDog function between CI858 and NPCT. 0 = disabled, 1 = enabled (default) |
| 3 | CALC_INT | in | bool | Speed calculation interval. 0 = 250 us (=default), 1 = 10 ms |
| 4 | MODE1 | in | dword | Encoder channel 1 configuration command. Default = 17 (HEX) |
| 5 | MODE2 | in | dword | Encoder channel 2 configuration command. Default = 17 (HEX) |
| 6 | PPR1 | in | int | Encoder channel 1 pulse numbers. |
| 7 | PPR2 | in | int | Encoder channel 2 pulse numbers. |
| 8 | BOARD_St | out | dword | Board configuration status. See next table. |
| 9 | CH1Conf_St | out | dword | Encoder channel 1 configuration status. |
| 10 | CH2Conf_St | out | dword | Encoder channel 2 configuration status. |
| 11 | ERR | out | bool | Communication status. 1 = Error. See *Output Error* on page 82. |
| 12 | Outword2_ds3 | out | dint | Second output word of DS 3 |
| 13 | Outword3_ds3 | out | dint | Third output word of DS 3 |
| 14 | Outword2_ds4 | out | dint | Second output word of DS 4 |
| 15 | Outword3_ds4 | out | dint | Third output word of DS 4 |
| 16 | Outword3_ds5 | out | dint | Second output word of DS 3 |
| 17 | Inword3_ds3 | in | dint | Third input word of DS 3 |
| 18 | Inword3_ds4 | in | dint | Third input word of DS 4 |
| 19 | Inword3_ds5 | in | dint | Third input word of DS 5 |
| 20 | UnitStatus | in | dint | Unit status |

Table 25. Basic configuration status of BOARD_St.

| Bit | Name | Description |
|---|---|---|
| D0 | WD_TO | WatchDog time out: 0 = 100 ms, 1 = 1 s |
| D1 | WD_ENA | WatchDog enable: 0 = disabled, 1 = enabled |
| D2 | not used | |
| D3 | CALC_INT | Speed calculation interval: 0 = 250 us, 1 = 10 ms |
| D4 | CTRL_TYPE | Application controller type: 0 = CI858, 1 = AMC |
| D5 | CPLD_CONF | CPLD configuration status:<br>0 = configuration completed,1 = configuration in progress |
| D6 | CPLD_ERR | Error in configuring the CPLD:<br>0 = configuration successful, 1 = configuration error |

Table 26. Encoder channel configuration commands of MODE1, MODE2.

| Bit | Name | Description |
|---|---|---|
| D0 | EOSC_MODE | Selection of edge oscillation inhibition mode: 0 = disabled,1 = enabled<br>The edge oscillation inhibition mode, EOSC_MODE, is used to prevent false occurrences of pulse edges that are caused mainly by mechanical vibration of the motor shaft at slow or zero speed. When this mode is enabled, pulse edges are rejected unless they occur alternately on CHA and CHB, i.e. an edge is rejected if the previous edge occurred on the same channel. Both CHA and CHB must be connected when EOSC_MODE is enabled. |
| D1...D2 | EDGE_MODE | Selection of pulse counter edge mode:<br>00 = Rising edges of CHA are counted, CHB determines the direction.<br>01 = Rising and falling edges of CHA are counted, CHB is not used.<br>10 = Rising and falling edges of CHA are counted, CHB determines the direction.<br>11 = Rising and falling edges of CHA and CHB are counted, CHB determines the direction. |
| D3 | POS_MODE | Selection of position value calculation mode: 0 = incremental mode,<br>1 = absolute mode<br>In incremental mode the position value varies between -32768...32767. Function Block POS_IL1 extends the 16-bit position value into a 32-bit value for wider range measurements.<br>In absolute mode the position value varies between 0... (edgeno * ppr)-1 indicating the angular displacement from the zero position, i.e. the encoder zero pulse. The absolute mode position calculation requires a rising edge of the zero pulse to be selected as the position strobe signal. |
| D4 | SYNC_MODE | Selection of channel synchronisation mode: 0 = disabled, 1 = enabled<br>In synchronised speed measurement mode the speed and position measurement and the DDCS communication are tied together in a way that ensures that the speed and position values of both encoder inputs (CH1 and CH2) are measured during the same cycle time. |
| D5...D7 | not used | |

Table 27. Encoder channel configuration status of CH1Conf_St, CH2Conf_St

| Bit | Name | Description |
|---|---|---|
| D0 | CH_CONF | Channel configuration status:<br>0 = channel is not configured, default values are in use<br>1 = channel is configured |
| D1 | ZERO_STAT | Indicates detection of a single zero strobe*:<br>0 = zero strobe not detected, 1 = zero strobe detected<br>ZERO_STAT is set when the first zero strobe is detected. For dynamic monitoring of zero strobe occurrences bit NEW_ZERO can be used. |
| D2 | NEW_SPEED | Indicates that a new speed value is available:<br>0 = not available, 1 = available<br>NEW_SPEED is set when the calculated speed is based on actual pulses i.e. at least one pulse is received during the last sampling time. If calculated speed is an estimate, NEW_SPEED is not set. |
| D3 | ENC_STAT | Encoder motion status: 0 = running, 1 = stopped |
| D4...D5 | not used | |
| D6 | NEW_ZERO | Indicates detection of a new zero strobe:<br>0 = not detected. 1 = detected.<br>NEW_ZERO is set if a zero strobe has been detected since the last communication request. |
| D7 | SPEED_SIGN | Sign of the current speed value: 0 = positive, 1 = negative |

*The position value can be set to zero with an external signal, zero strobe. The zero strobe signal can be a combination of the encoder zero pulse CHZ and a digital input.

**SPEEDPO1**

Speed and position measurement values of encoder channels CH1 and CH2 are read from the NPCT-01 board with function block SPEEDPO1.

The actual selection of the I/O unit is made in the Control Builder's hardware editor by connecting respective channels of the NPCT board to input and output signals 16...26 (outword2_ds1... UnitStatus). See *Table 28* on page 70 and *Connections* on page 44.

Table 28. Input and output signals of the SPEEDPO1.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | FILT1 | in | dint | Sampling time of the speed calculation (CH1). Input signal CALC_INT of PCT_CONF effects the sampling time.<br>If the calculation interval is set to 10 ms, CALC_INT = 1, one sample time unit corresponds to 40 ms.<br>If CALC_INT = 0, 1...25 = 1...25 ms (default 2 ms)<br>If CALC_INT = 1, 1...25 = 40...1000 ms |
| 2 | STRB1 | in | dword | Zero strobe configuration command (CH1). STRB1 is set with a rising edge of WR_STRB1. (0 -> 1). See next table. |
| 3 | WR_STRB1 | in | bool | Dynamic input for new zero strobe configuration (CH1). STRB1 is set with a rising edge of WR_STRB1 (0 -> 1). |
| 4 | FILT2 | in | dint | Sampling time of speed calculation (ch2). Input signal CALC_INT of PCT_CONF effects the sampling time.<br>If the calculation interval is set to 10 ms (CALC_INT=1) one sample time unit corresponds to 40 ms.<br>If CALC_INT=0, 1...25 = 1...25 ms, (default 2 ms)<br>If CALC_INT=1, 1...25 = 40...1000 ms |
| 5 | STRB2 | in | dword | Zero strobe configuration command (CH2). STRB2 is set with a rising edge of WR_STRB2. (0 -> 1). See next table. |
| 6 | WR_STRB2 | in | bool | Dynamic input for new zero strobe configuration (CH2). STRB2 is set with a rising edge of WR_STRB2. (0 -> 1) |
| 7 | SPEED1 | out | dint | Actual speed value (CH1). |
| 8 | POSACT1 | out | int | Actual position value (CH1). |
| 9 | STRBRDY1 | out | bool | Zero strobe detection (CH1). 1 = detected |
| 10 | STATUS1 | out | dword | Encoder channel 1 status. See table STATUS. |
| 11 | SPEED2 | out | dint | Actual speed value (CH2). |
| 12 | POSACT2 | out | int | Actual position value (CH2). |
| 13 | STRBRDY2 | out | bool | Zero strobe detection (CH2). 1 = detected |
| 14 | STATUS2 | out | dword | Encoder channel CH2 status. See table STATUS |
| 15 | ERR | out | bool | Communication status. 1 = Error<br>See *Output Error* on page 82. |
| 16 | Outword2_ds1 | out | dint | Second output word of DS 1. |

Table 28. Input and output signals of the SPEEDPO1.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 17 | Outword3_ds1 | out | dint | Third output word of DS 1. |
| 18 | Outword2_ds2 | out | dint | Second output word of DS 2. |
| 19 | Outword3_ds2 | out | dint | Third output word of DS 2. |
| 20 | Inword1_ds1 | in | dint | First input word of DS 1. |
| 21 | Inword2_ds1 | in | dint | Second input word of DS 1. |
| 22 | Inword3_ds1 | in | dint | Third input word of DS 1. |
| 23 | Inword1_ds2 | in | dint | First input word of DS 2. |
| 24 | Inword2_ds2 | in | dint | Second input word of DS 2. |
| 25 | Inword3_ds2 | in | dint | Third input word of DS 2. |
| 26 | UnitStatus | in | dint | Unit status |

Value of POSACT1/POSACT2 is equal to the amount of pulse edges received from the incremental encoder, i.e. if four encoder signal edges are counted, the position value will increase by four times the encoder pulse number with each turn.

SPEED1/SPEED2 corresponds to the signed actual value of the tacho frequency. The frequency is calculated from the ratio of the number of received encoder pulse edges (Δp) and the time between the first and last received pulse edge (Δt) during the selected sampling time FILT1/FILT2.

*Scaling of SPEED1:*

When tacho frequency is 500000 Hz the output value of SPEED1 is 8388607.

SPEED1 = (Δp * 8388607) / (Δt * 500000 * n)
n is the number of edges counted (1, 2 or 4)

*Calculation example:*

$f_{encoder}$ = 25000 Hz (four edges = 100000 Hz)
Δt = 0.016 ms
Δp = (0.016 ms / 1 s) * 100000 = 1600
SPEED1 = (1600 * 8388607) / (0.016 * 500000 * 4) = 419430.35

Table 29. Zero strobe configuration commands STRB1, STRB2

| Bit | Name | Description |
|---|---|---|
| D0...D2 | ZERO_MODE | Selection of zero strobe* mode for encoder channel 1/2:<br>000 = strobe disabled<br>001 = rising edge of the zero pulse<br>010 = rising edge of the zero pulse when DI1/DI2 = 0<br>011 = rising edge of the zero pulse when DI1/DI2 = 1<br>100 = rising edge of DI1/DI2 signal followed by a zero pulse<br>101 = DI1/2 falling edge followed by a zero pulse rising edge<br>110 = DI1/2 rising edge<br>111 = DI1/2 falling edge<br>DI1 is used for encoder channel 1 and DI2 for encoder channel 2. |
| D3 | ZERO_RST | Selection of zero strobe mode reset control:<br>0 = no reset, 1 = reset zero strobe mode to 000 after zero strobe.<br>Setting ZERO_RST to 1 makes it possible to accept a single zero strobe. A new zero strobe is enabled by writing a new value to ZERO_MODE and setting bit ZERO_UPD to 1. |
| D4 | ZERO_FILT | Selection of hardware filter time constant for DI1/2: 0 = 100 us, 1 = 5 ms |
| D5 | ZERO_UPD | Updating of zero strobe configuration: 0 = not updated,<br>1 = the zero strobe configuration is updated according to bits D0...D4. |
| D6...D7 | not used | |

* Position value can be set to zero with an external hardware signal, a zero strobe. The zero strobe signal can be a combination of the encoder zero pulse CHZ and a digital input, see table *Table 27* on page 69. Status signal NEW_ZERO indicates a detection of a new zero strobe. NEW_ZERO is set if a zero strobe has been detected since the last communication request.

Table 30. Encoder channel status: STATUS1, STATUS2

| Bit | Name | Description |
|---|---|---|
| D0 | CH_CONF | Channel configuration status:<br>0 = not configured, default values are in use, 1 = configured |
| D1 | ZERO_STAT | Indicates detection of a single zero strobe:<br>0 = not detected, 1 = detected<br>ZERO_STAT is set when the first zero strobe is detected. NEW_ZERO can be used for dynamic monitoring of zero strobe occurrences. |
| D2 | NEW_SPEED | Indicates if the new speed value is available:<br>0 = not available, 1 = available<br>NEW_SPEED is set when the calculated speed is based on actual pulses i.e. at least one pulse is received during the last sampling time. If calculated speed is an estimate, NEW_SPEED is not set. |
| D3 | ENC_STAT | Encoder motion status: 0 = running, 1 = stopped |
| D4...D5 | not used | |
| D6 | NEW_ZERO | Indicates detection of a new zero strobe:<br>0 = not detected, 1 = detected<br>NEW_ZERO is set if a zero strobe has been detected since the last request. |
| D7 | SPEED_SIGN | Sign of the current speed value: 0 = positive, 1 = negative |

**SPEED_R**

Function block SPEED_R converts long integer (IL) values of the actual speed into real number values (rpm). SPEED_R is intended for use with function blocks SPEEDPO1 and TAC_SP.

Table 31. Input and output signals of the SPEED_R.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | PPR | in | real | Encoder channel pulse number. |
| 2 | NSPEED | in | real | Nominal rotational speed in rpm. |
| 3 | SCALE | in | real | Output value at nominal rotational speed in rpm. |
| 4 | SPEED | in | dint | Actual speed. |
| | RPM | out | real | Actual speed in rpm. |

SPEED_R realises the following formula:

RPM = (60 * 500000 * SPEED * SCALE) / (PPR * 8388607 * NSPEED)

Conversion to rpm is made by using the following part of the formula:

RPM = (60 * 500000 * SPEED) / PPR * 8388607

Scaling of rpm value is made by SCALE and NSPEED.

*Example:*

PPR = 2048
SPEED = 419430
SCALE = 1500
NSPEED = 1500 rpm
RPM = (60 *500000 * 419430 * 1500) / (2048 * 8388607 * 1500) = 732.421 rpm

**POS_IL1**

Function block POS_IL1 extends the 16-bit position value into a 32-bit value for wider range measurements. Function block POS_IL1 is intended to be used with function blocks SPEEDPO1 and TAC_SP.

Table 32. Input and output signals of POS_IL1.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | SW_STRB | in | bool | Zero strobe within the application program. (0 -> 1 new zero strobe) |
| 2 | STRB_VAL | in | dint | Value of the zero strobe. |
| 3 | POSACT | in | int | 16-bit actual position value. |
| 4 | STATUS SPEED | in | dword | Encoder channel status. |
| 5 | POS_IL | out | dint | 32-bit actual position value. |

POSACT is a 16-bit position value received from the output of SPEEDPO1 or TAC_SP. STATUS SPEED indicates the encoder channel status, which is received from the output of SPEEDPO1.

When zero strobe is detected (bit 6 of STATUS signal of function block SPEEDO1 is 1), POS_IL is set to the value of STRB_VAL. A zero strobe can also be activated from the application program by connecting a rising edge to the input SW_STRB.

Incremental mode should be selected for position calculation with function blocks SPEEDPO1 or TAC_SP.

**CONV24**

AC 800M dataset communication consists of 16-bit data words while NPCT communication consists of 24-bit data words. To simplify the programming of the NPCT board, function block CONV24 is used to convert 24-bit input values INVAL1 and INVAL2 into tree 16-bit output words outword1...3 (or inword1..3 into OUTVAL1 and OUTVAL2).

Table 33. Input and output signals of CONV24.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | INVAL1 | out | dint | 24-bit input value. |
| 2 | INVAL2 | out | dint | 24-bit input value. |
| 3 | OUTVAL1 | in | dint | 24-bit output value. |
| 4 | OUTVAL2 | in | dint | 24-bit output value. |
| 5 | ERR | out | bool | Communication status. 1 = Error.<br>See *Output Error* on page 82. |
| 6 | Outword1 | out | dint | First output word of DS x.* |
| 7 | Outword2 | out | dint | Second output word of DS x.* |
| 8 | Outword3 | out | dint | Third output word of DS x.* |
| 9 | Inword1 | in | dint | First input word of DS x.* |
| 10 | Inword2 | in | dint | Second input word of DS x.* |
| 11 | inword3 | in | dint | Third input word of DS x.* |
| 12 | UnitSatus | in | dint | Unit status |

* x = 6...10

## NTAC

NTAC-02 is a speed and position measurement unit which can be used with CI858. Software revision of the NTAC-02 unit should be 2.07 or later (absolute mode for position calculation does not work correctly with older revisions). For more information see *NTAC-0x/NDIO-0x/NAIO-0x Installation and Start-up Guide [3AFY 58919730 (English)]*.

**Note:** NTAC unit functions only with original configuration settings i.e. when dataset 2 is OFF. This limitation is due to the fact, that all dataset communications from AC 800M to Special I/O units must be cyclic. If NTAC-02 receives a configuration dataset, it aborts current operation and restarts all internal counters. (it is not possible to send datasets “on demand"-basis)

There is one function block, TAC_SP, for handling the NTAC-02 unit.

### TAC_SP

The encoder speed and position measurement values are read from NTAC-02 with function block TAC_SP. Functions SPEED_R and POS_IL1 can be used with TAC_SP. See *SPEED_R* on page 73 and *POS_IL1* on page 74.

The actual selection of the I/O unit is made in the Control Builder’s hardware editor by connecting respective channels of the NTAC unit to input signals 20...26 (inword1... UnitStatus). See *Table 34* on page 77 and *Connections* on page 44.

Table 34. Input and output signals of the TAC_SP.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 7 | SPEED | out | dint | Actual speed value |
| 8 | POSACT | out | int | Actual position value |
| 9 | NEW_SP | out | bool | Indicates that new speed value is available.<br>1 = detected |
| 10 | ENC_STAT | out | bool | Encoder motion status.<br>0 = running, 1 = stopped |
| 15 | ERR | out | bool | Communication status. 1 = Error<br>See *Output Error* on page 82. |
| 20 | Inword1 | in | dint | First input word of DS 1. |
| 21 | Inword2 | in | dint | Second input word of DS 1. |
| 22 | Inword3 | in | dint | Third input word of DS 1. |
| 26 | UnitStatus | in | dint | Unit status. |

SPEED corresponds to the signed actual value of the tacho frequency. The frequency is calculated from the ratio of the number of received encoder pulse edges (Δp) and the time between the first and last received pulse edge (Δt) during the selected sampling time FILT.

*Output SPEED scaling:*

When tacho frequency is 500000 Hz then output value of SPEED is 8388607.

SPEED = (Δp * 8388607) / (Δt * 500000 * n)
n is the number of edges counted (1, 2 or 4).

*Calculation example:*

$F_{encoder}$ = 25000 Hz (four edges, 100000 Hz)
Δt = 0.016 ms
Δp = (0.016 ms / 1 s) * 100000 = 1600
SPEED = (1600 * 8388607) / (0.016 * 500000 * 4) = 419430.35

Value of POSACT is equal to the amount of pulse edges received from the incremental encoder, i.e. if four encoder signal edges are counted, the position value will change by four times the encoder pulse number with each turn.

## NWIO

NWIO-01 Watchdog & I/O is a digital I/O option unit. It contains a watchdog logic for supervising the operation of the AC 800M and the CI858 software. The unit has two digital inputs (24 VDC or 115/230 VAC) and two digital outputs. Using a slide switch on the top of the unit, the watchdog logic can be connected to control the state of the digital output DO2.

The watchdog supervises the following operations of the CI858:

- Functioning of the CI858 application program.
- Communication over the dual port memory.
- Functioning of the CI858 operative system.
- Communication on the optical I/O ring.

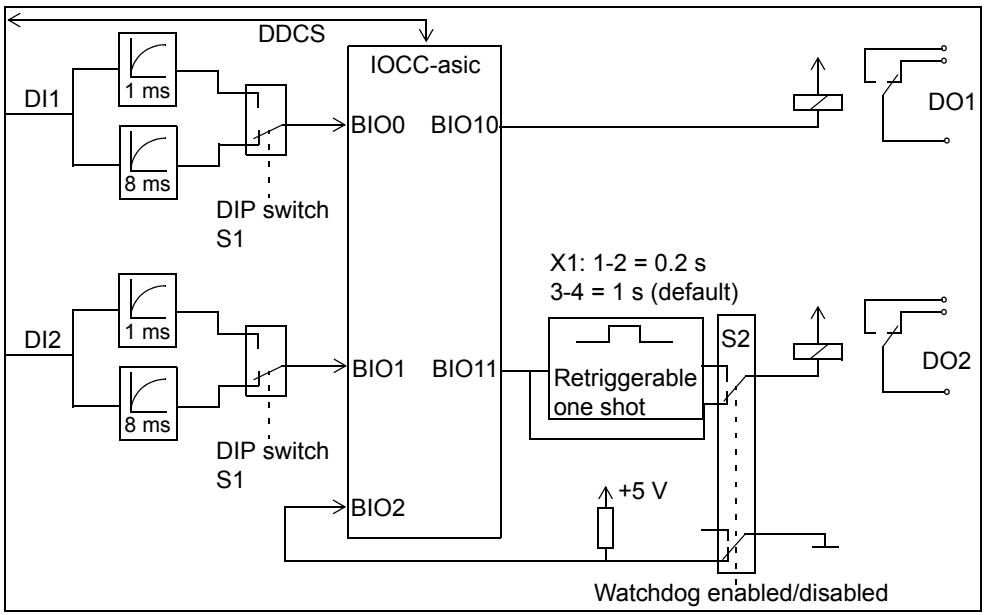



Figure 16. Simplified block diagram of the NWIO-01 unit.

There are two function blocks for handling the NWIO-01 unit: NWIO1DI and NWIO1DO.

**NWIO1DI**

Function block NWIO1DI is used for reading digital input signals from the NWIO-01 unit.

The actual selection of the I/O unit is made in the Control Builder's hardware editor by connecting respective channels of the NWIO unit to input signals 5...8 (inword1... UnitStatus). See *Table 35* on page 80 and *Connections* on page 44.

Table 35. Input and output signals of NWIO1DI.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | DI1 | out | bool | Output for digital input DI1. |
| 2 | DI2 | out | bool | Output for digital input DI2. |
| 3 | WD_STAT | out | bool | Selection of WatchDog function.<br>0 = not selected, 1 = selected |
| 4 | ERR | out | bool | Communication status. 1 = Error<br>See *Output Error* on page 82. |
| 5 | Inword1 | in | dint | First input word of DS 100 (R). |
| 6 | Inword2 | in | dint | Second input word of DS 100 (R). |
| 7 | Inword3 | in | dint | Third input word of DS 100 (R). |
| 8 | UnitStatus | in | dint | Unit status. |

**NWIO1DO**

Function block NWIO1DO is used for writing digital output signals to the NWIO-01 unit.

The actual selection of the I/O unit is made in the Control Builder's hardware editor by connecting respective channels of the NDIO unit to input and output signals 5...9 (outword1... UnitStatus). See *Table 36* on page 81 and *Connections* on page 44.

Table 36. Input and output signals of NWIO1DO.

| No | Name | Direction | Data type | Description |
|---|---|---|---|---|
| 1 | DO1 | in | bool | Input for digital output DO1 |
| 2 | DO2 | in | bool | Input for digital output DO2. |
| 3 | WD_SEL | in | bool | Selection of the DO2 operation mode.<br>DO2 can be used as normal software controlled digital output or as a watchdog output. |
| 4 | ERR | out | bool | Communication status.<br>See *Output Error* on page 82. |
| 5 | outword1 | out | dint | First output word of DS 100 (R/W). |
| 6 | outword2 | out | dint | Second output word of DS 100 (R/W). |
| 7 | outword3 | out | dint | Third output word of DS 100 (R/W). |
| 8 | inword1_dummy | in | dint | First input word (=dummy) of DS 100 (R/W). |
| 9 | UnitStatus | in | dint | Unit status |

Digital output DO2 can be used as a normal software controlled digital output or as a watchdog output. The operation mode is selected with the WATCHDOG ENABLED / DISABLED switch on the top of the unit and with input signal WD_SEL. In watchdog mode, a retriggerable one shot circuit is connected between the software controlled output and the actual output relay. Changing the state of digital output DO2 refreshes the one shot circuit (software oscillator is made by the inverter element).

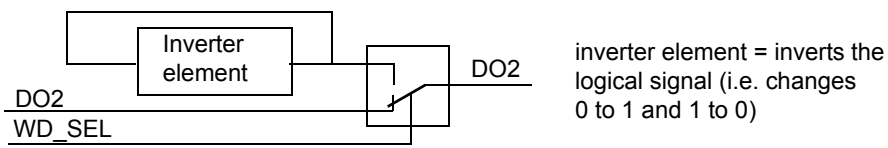



Figure 17. Watchdog operation.

The factory setting of the watchdog time constant is 1 s but it can be shortened to 0.2 s with jumper X1 inside the unit (X1, pins 1-2 = 0.2 s; pins 3-4 = 1 s).

The cycle time of the oscillator is 2 * execution interval of the NWIO1DO. The cycle time must be fast enough compared with the time constant selection in the NWIO-01 unit.

Time constant 1 s: Execution interval of the task ≤ 100 ms, thus the cycle time of the oscillator is ≤ 200 ms.

Time constant 0,2 s: Execution interval of the task ≤ 20 ms, thus the cycle time of the oscillator is ≤ 40 ms.

## NCTI

For information about the NCTI-01 Crane Transducer Interface see *Crane transducer Interface User's manual [3ASC23F207 (English)]*.

## NDSC

NDSC-01 is the control board of the ACS600 Diode Supply Unit. For more information see *ACA 631/633 Diode Supply Section User's Manual [3AFY 61451544 (English)].*

Configuration of the NDSC board is similar to the configuration of a drive.

## Output Error

When a communication error is detected in the cyclic DDCS message, output ERR is set according to the UnitStatus signal. Output ERR is cleared when an acknowledgement to the cyclic DDCS message without any errors is received from the I/O unit and old error indications are acknowledged.

Table 37. Output error (derived from UnitStatus).

| Unit status | Error | Description |
|---|---|---|
| 0 | 0 | No errors detected. |
| 1 | 1 | Communication fails |
| 2 | 1 | Unacknowledged alarm or forced operation (i.e. output signal is forced to a value defined by the user in the hardware editor) |

# Changing the CI858 System Settings

When a modified hardware configuration is downloaded to the controller, communication with hardware units can be interrupted:

- If modified CI858 parameters are downloaded to the controller, the CI858 communication is interrupted and the CI858 will reboot.
- If modified drive parameters are downloaded to the controller, communication with the drive is interrupted and a drive fault message indicating communication loss might be activated. If BusManager is not selected to monitor the connection, the fault can be avoided by adjusting the time delay of the drive communication loss supervision.
- If modified I/O parameters are downloaded to the controller, communication with the I/O unit is interrupted.
- If a drive or an I/O is added to or deleted from the hardware tree and the changes are downloaded to the controller, the CI858 will reboot.
- If the hardware tree positions of different types of drives or I/O's are changed and the changes are downloaded to the controller, the CI858 will reboot. Changing the position between two similar units will not result in CI858 rebooting.
- Changing the connected channels of a drive or an I/O causes recalculation of the connections.

# Online Help

For more information on how to configure the CI858 unit with the Control Builder Engineering Tool, see Online Help.

# Appendix A Technical Data

## General Technical Data

Table 38. General technical data

| | |
|---|---|
| Protection class: | IP20 |
| Approval: | CE-marked and meets the requirements specified in EMC Directive 89/336/EEC according to the standards EN 50081-2 and EN 61000-6-2. |

## Ambient Data

Table 39. Ambient conditions

| | **Operation** installed for stationary use | **Storage** in a protective package |
|---|---|---|
| Air temperature | +5 to +55 °C | -25 to +70 °C |
| Relative humidity (non-consending) | 5 to 95% | 5 to 95% |

## LED Indicators

Table 40. LED indicators

| LED marking | Color | Reset state | Normal state | Description |
|---|---|---|---|---|
| F | RED | ON | OFF | Unit error or unit initializing. |
| R | GREEN | OFF | ON | Unit running. |
| Dbus; Rx | YELLOW | | | RECIEVE LED indicating reception of data frames on the DriveBus. |
| DBus; Tx | YELLOW | | | TRANSMIT LED indicating transmission of data frames on the DriveBus. |
| I/O; Rx | YELLOW | | | RECIEVE LED indicating reception of data frames on the I/O bus. |
| I/O; Tx | YELLOW | | | TRANSMIT LED indicating transmission of data frames on the I/O bus. |
| Tools | YELLOW | | | RECIEVE LED indicating reception of data frames on the PC Tool Bus. * |
| Tools | YELLOW | | | TRANSMIT LED indicating transmission of frames on the PC tool bus.* |

*This LED is not visible in the front of the unit. It can be seen through the ventilation holes in the plastic housing.

## Power Supply Requirements

Power supply for the unit is taken from L+ and L- on the CEX bus.

Table 41. Technical data for the power supply.

| | Minimum | Typical | Maximum |
|---|---|---|---|
| external supply voltage | 17 V | 24 V | 30 V |
| external supply current | | | 200 mA |

## Optical Connectors

The CEM board connects to the drive system via three optical receiver/transmitter pairs. HP/Agilent Technologies Versatile Link Series (HFBR family) optical receiver/transmitters are used. Transmission speed of the optical fibres is 4 Mbit/s.

Table 42. Optical connectors D27-D29.

| Pin | Designator | Description | Pin | Designator | Description |
|---|---|---|---|---|---|
| 1 | TxD | transmit data | 2 | Rxd | receive data |

## DIN rail - TS 35/7.5

The figure below shows the profile and dimensions in mm (in.) of the DIN-rail TS 35/7.5.

Normal supplied length is 2m (6.6ft.) (to standard DIN 46 277, EN 50 022).

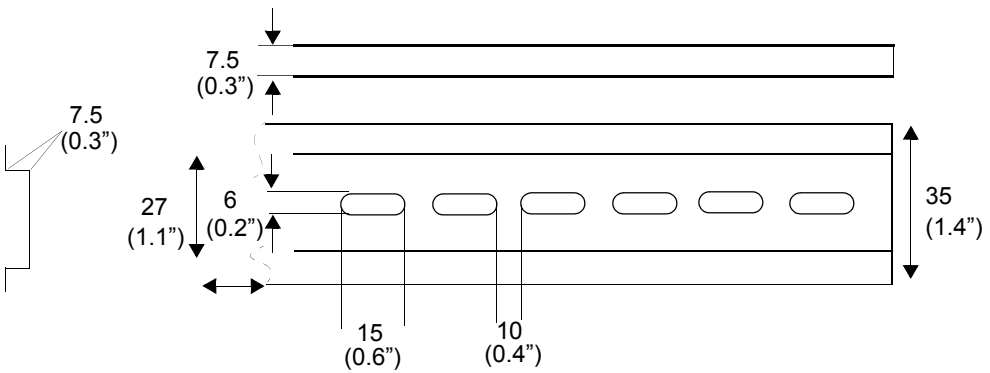



Figure 18. DIN-rail TS 35/7.5 dimensions

## CEB Connector, X1

The communication expansion module board CEM connects to the CEX bus via connector X1.

Table 43. CEB connector X1.

| Pin | Designator | Description | Pin | Designator | Description |
|---|---|---|---|---|---|
| A1 | CEX_L- | power supply | B1 | CEX_L- | power supply |
| A2 | CEX_L+_PRGS | power supply | B2 | CEX_L+ | power supply |
| A4 | 0 V | | B4 | INSERTED_HI_N | used in power supply |
| A5 | 0 V | | B5 | 0 V | |
| A6 | CLK+ | CEX bus clock | B6 | 0 V | |
| A7 | CLK- | CEX bus clock | B7 | 0 V | |
| A8 | 0 V | | B8 | 0 V | |
| A9 | DATA+ | CEX bus data | B9 | 0 V | |
| A10 | DATA- | CEX bus data | B10 | 0 V | |
| A11 | 0 V | | B11 | 0 V | |
| A12 | STRB+ | CEX bus data strobe | B12 | 0 V | |
| A13 | STRB- | CEX bus data strobe | B13 | 0 V | |
| A14 | 0 V | | B14 | 0 V | |
| A15 | INT_N | CEX bus interrupt | B15 | 0 V | |
| A16 | 0 V | | B16 | GSA | |
| A17 | INIT | CEX bus reset | B17 | GSB | |
| A18 | 0 V | | B18 | 0 V | |
| A19-A37 | not used | not used | B19-B24 | not used | not used |
| A38 | EM | | B25 | 0 V | |
| A39 | 0 V | | B26-B37 | not used | not used |

| Pin | Designator | Description | Pin | Designator | Description |
|---|---|---|---|---|---|
| A40 | EM | | B38 | EM | |
| A41 | CEX_L+_PRCG | power supply | B39 | INSERTER_LO_N | used in power supply |
| A42 | CEX_L- | power supply | B40 | EM | |
| | | | B41 | CEX_L+ | power supply |
| | | | B42 | CEX_L- | power supply |

## CEX Bus Interface

The CEX bus signals are listed in below.

Table 44. CEX bus signals.

| Signal name | Level | Description | Note |
|---|---|---|---|
| DATA+ | LVDS | Data | |
| DATA- | LVDS | Data | |
| STRB+ | LVDS | Data strobe | |
| STRB- | LVDS | Data strobe | |
| CLK+ | LVDS | CEX bus clock | |
| CLK- | LVDS | CEX bus clock | |
| INT0_N | TTL | Interrupt to CEX bus master | Open collector output |
| INIT | TTL | Init (reset) selected unit | |
| GSA(2:0) | TTL | Geographical select signals | |
| GSB(3:0) | TTL | Geographical select signals | |
| CEX_L+ | +24 V | Power supply | 17-30 V, 200 mA |
| CEX_L- | 0 V | Power supply | |

## Firmware Download

The CI858 firmware is downloaded with a special loading package.

For instructions on how to download the CI858 firmware, see Control$^{IT}$ for AC 800M CD-ROM.

ABB Oy
AC Drives
P.O. Box 184
FIN-00381 HELSINKI
FINLAND
Telephone  +358 10 22 211
Telefax  +358 10 22 22681
Internet http://www.abb.com

3AFE68237432 REV C/EN
EFFECTIVE 25.10.2004